MAR 22 1979

80-10017
CR-160342

TEXAS APPLICATIONS SYSTEM VERIFICATION AND TRANSFER
REMOTE SENSING INFORMATION SUBSYSTEM:

Functional Design

M. L. Brown, Jr.
A. M. Fails
M. V. Martin
A. S. Story
E. A. Weisblatt
Lockheed Electronics Company, Inc.
Systems and Services Division
1830 NASA Road 1
Houston, Texas 77058

NAS 9-15800
LEC-13258
JSC-14785
March 1979
Technical Report

Prepared for
EARTH OBSERVATIONS DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058

N80-13596

(E80-10017) TEXAS APPLICATIONS SYSTEM VERIFICATION AND TRANSFER REMOTE SENSING INFORMATION SUBSYSTEM: FUNCTIONAL DESIGN (Lockheed Electronics Co.) 405 p HC A18/MF A01 CSCL 05B G3/43

Unclas
00017

1. Report No.
JSC-14785

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
TEXAS APPLICATIONS SYSTEM VERIFICATION AND TRANSFER
REMOTE SENSING INFORMATION SUBSYSTEM:
Functional Design

5. Report Date
March 1979

6. Performing Organization Code

7. Author(s)
M. L. Brown, A. M. Fails, M. V. Martin, A. S. Story
E. A. Weisblatt

8. Performing Organization Report No.
LEC-13258

9. Performing Organization Name and Address
Lockheed Electronics Company, Inc.
Systems and Services Division
1830 NASA Road 1
Houston, Texas 77058

10. Work Unit No.

11. Contract or Grant No.
NAS 9-15800

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas 77058

13. Type of Report and Period Covered
Technical Report

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract
This document describes the functional design of the Remote Sensing Information Subsystem being developed as part of the Texas Natural Resources Information System by the National Aeronautics and Space Administration, Lyndon B. Johnson Space Center for the Texas Applications System Verification and Transfer program. The design contains all necessary functional modules required to provide an interactive processing system for both computer-assisted and manual processing.

PRECEDING PAGE BLANK NOT FILMED

17. Key Words (Suggested by Author(s))
remote sensing, Earth resources,
technology transfer

18. Distribution Statement

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

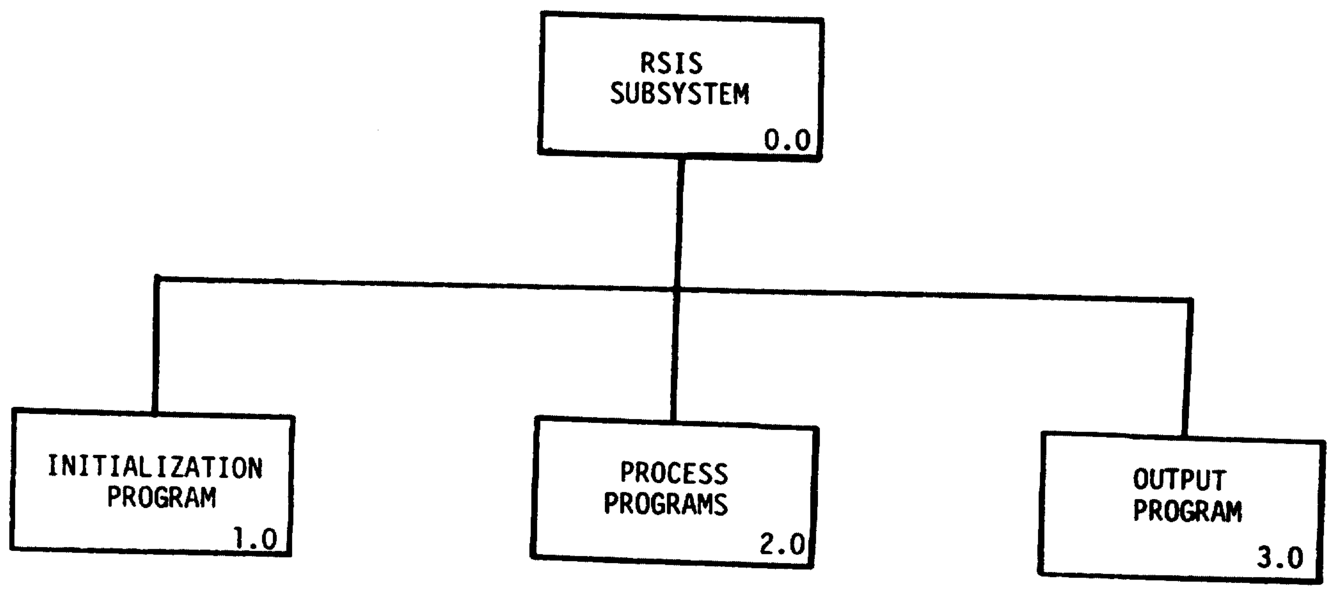
21. No. of Pages

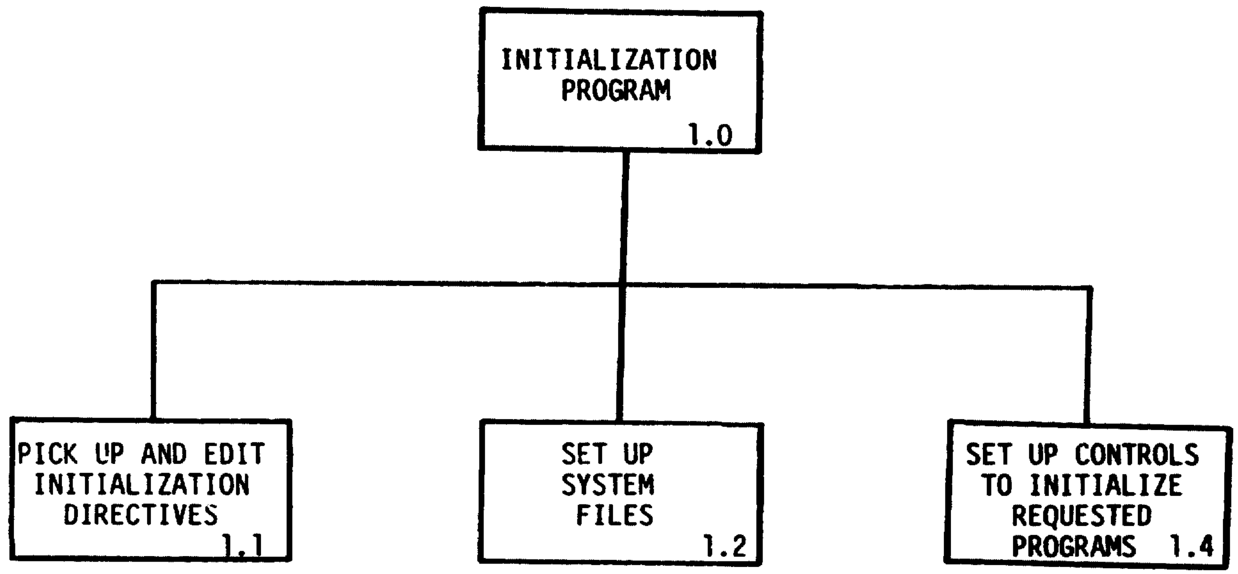
22. Price*

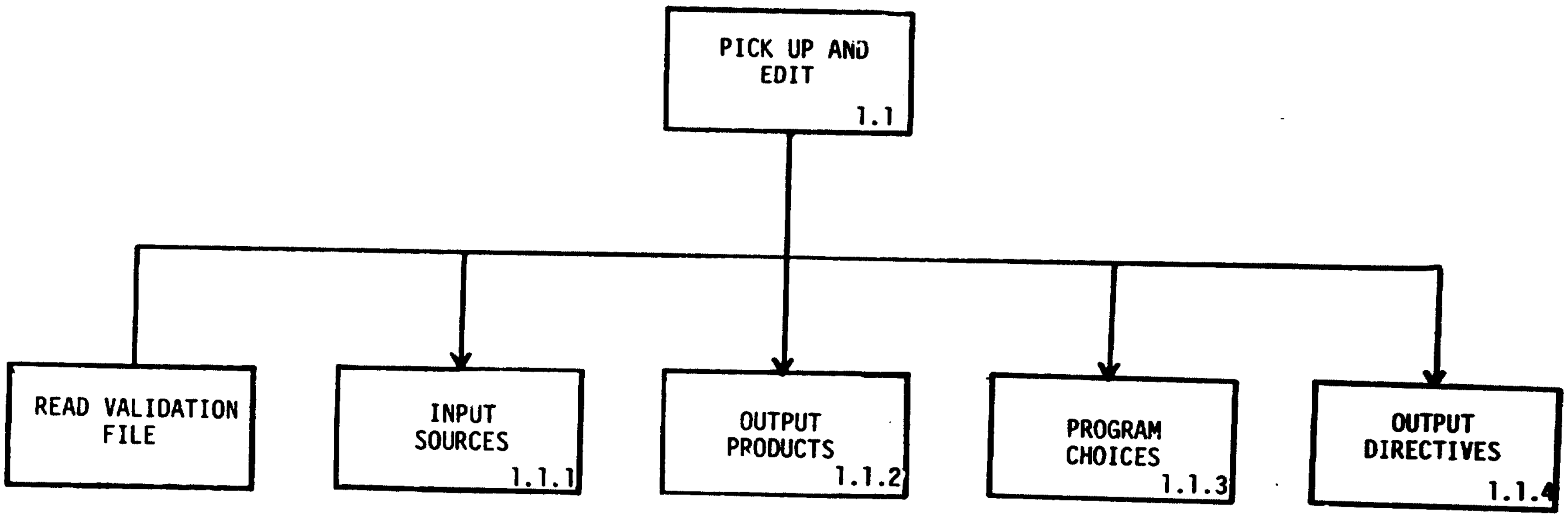
*For sale by the National Technical Information Service, Springfield, Virginia 22161

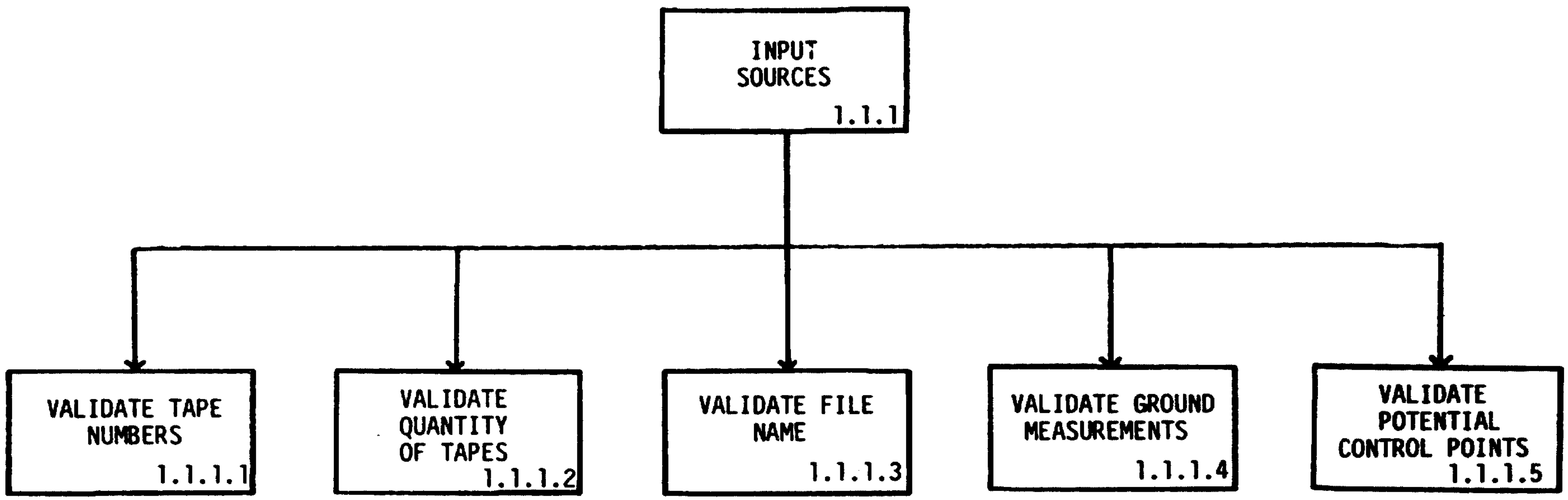
CONTENTS

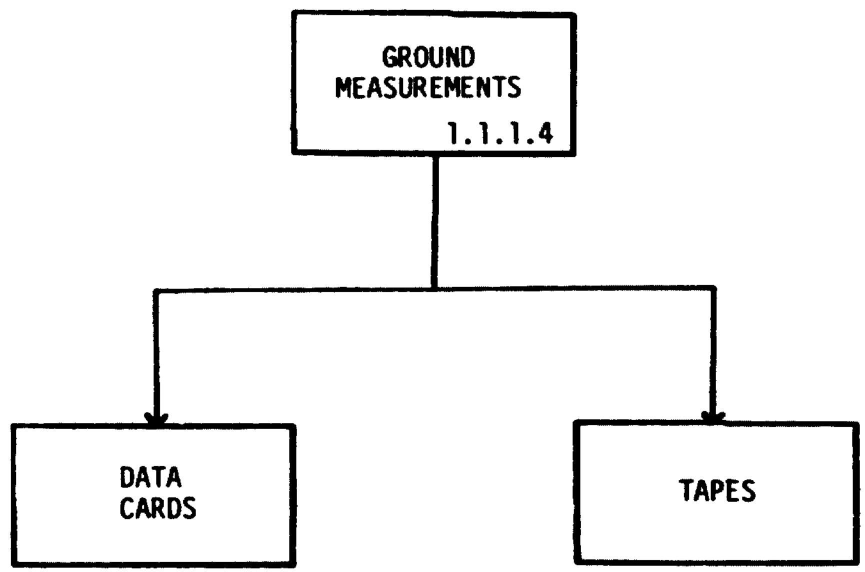
COMPUTER PROCESSING



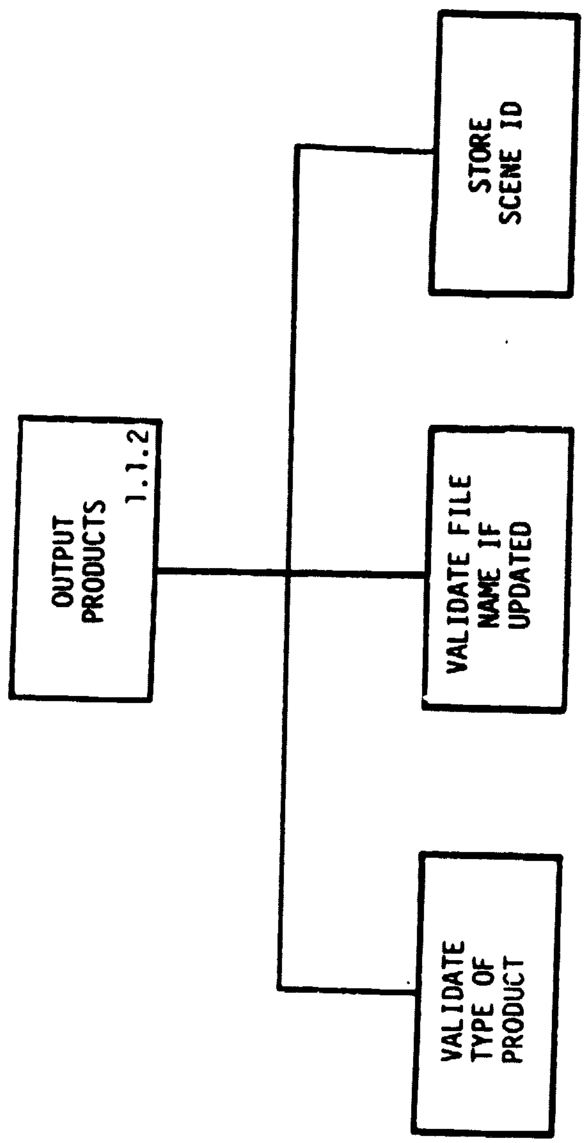

INITIALIZATION PROGRAM 1.0
PICK UP AND EDIT INITIALIZATION DIRECTIVES 1.1
SET UP SYSTEM FILES 1.2
SET UP CONTROLS TO INITIALIZE REQUESTED PROGRAMS 1.4

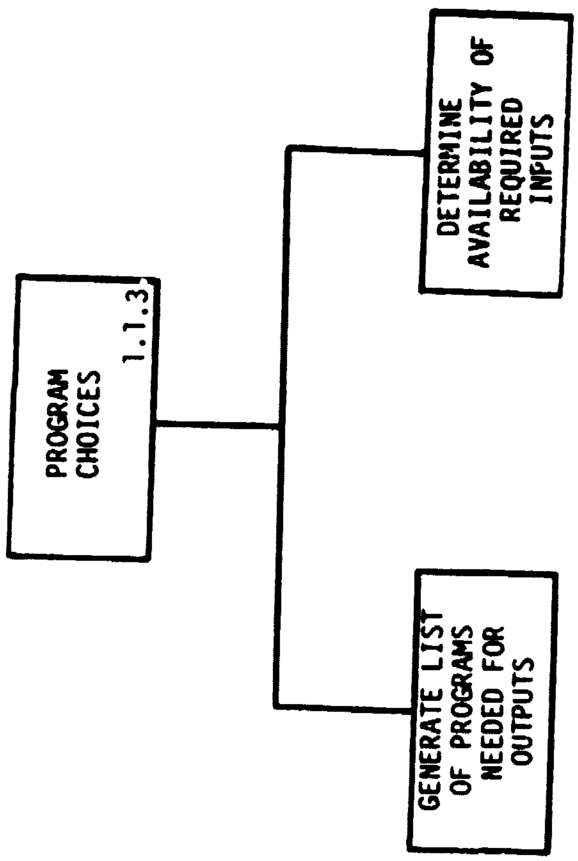

PICK UP AND EDIT
1.1
READ VALIDATION FILE
INPUT SOURCES
1.1.1
OUTPUT PRODUCTS
1.1.2
PROGRAM CHOICES
1.1.3
OUTPUT DIRECTIVES
1.1.4

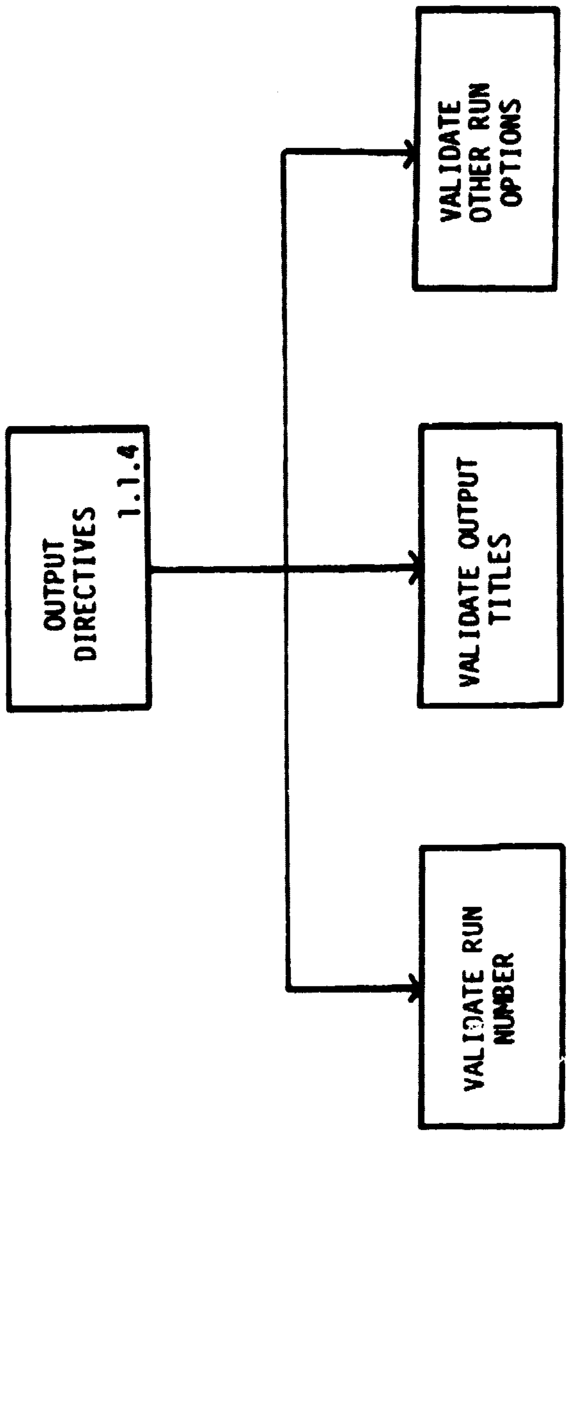

INPUT SOURCES 1.1.1
VALIDATE TAPE NUMBERS 1.1.1.1
VALIDATE QUANTITY OF TAPES 1.1.1.2
VALIDATE FILE NAME 1.1.1.3
VALIDATE GROUND MEASUREMENTS 1.1.1.4
VALIDATE POTENTIAL CONTROL POINTS 1.1.1.5

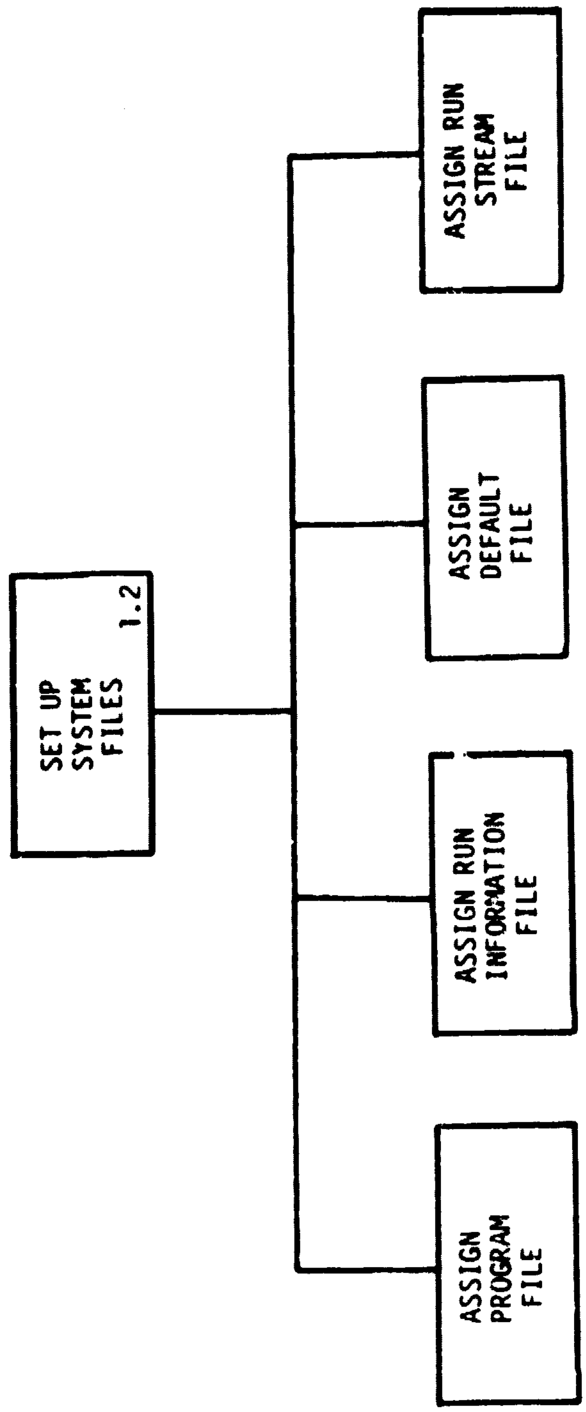


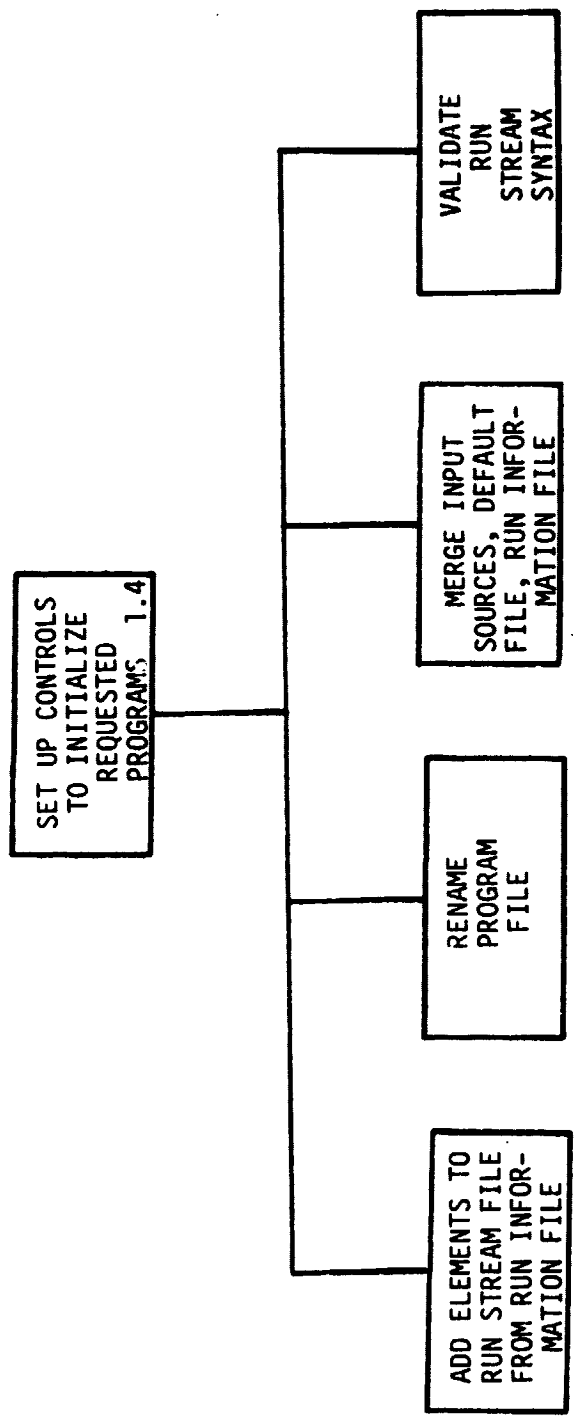
Note: Arrows indicate selective routing.

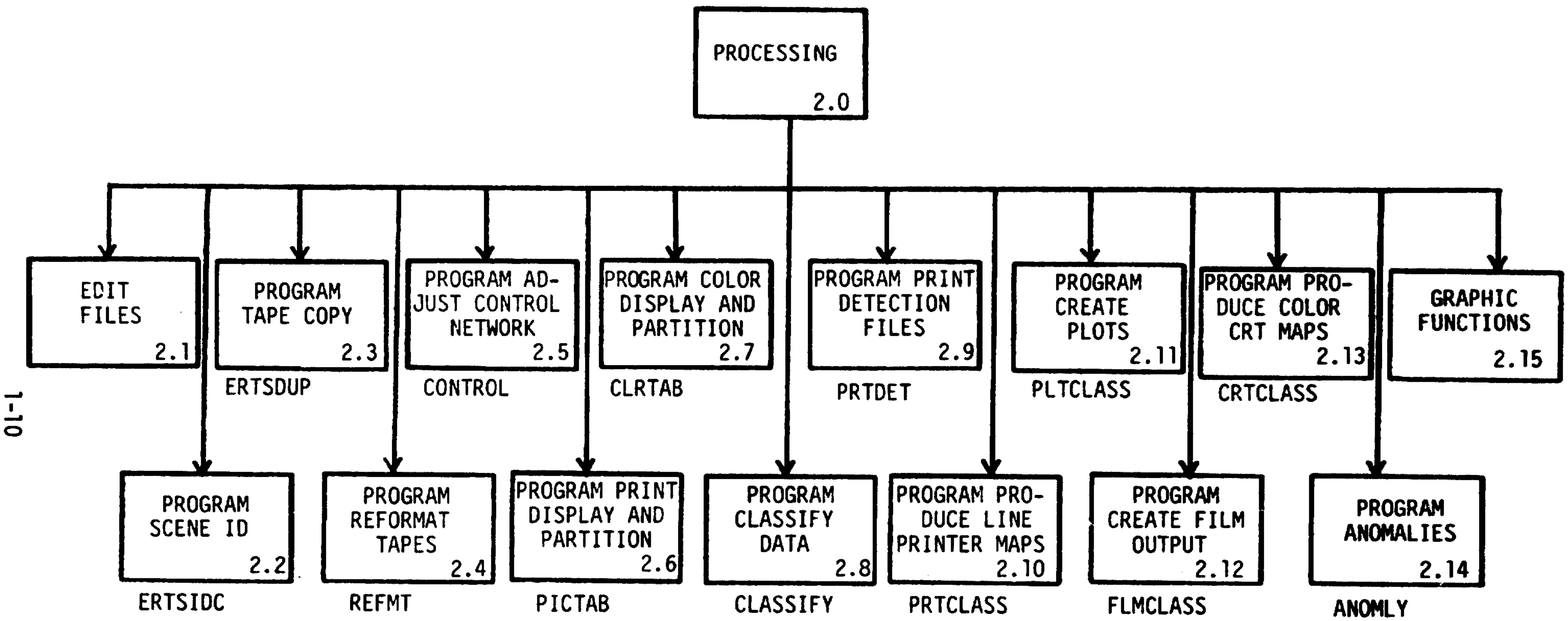

OUTPUT PRODUCTS
1.1.2
STORE SCENE ID
VALIDATE FILE NAME IF UPDATED
VALIDATE TYPE OF PRODUCT

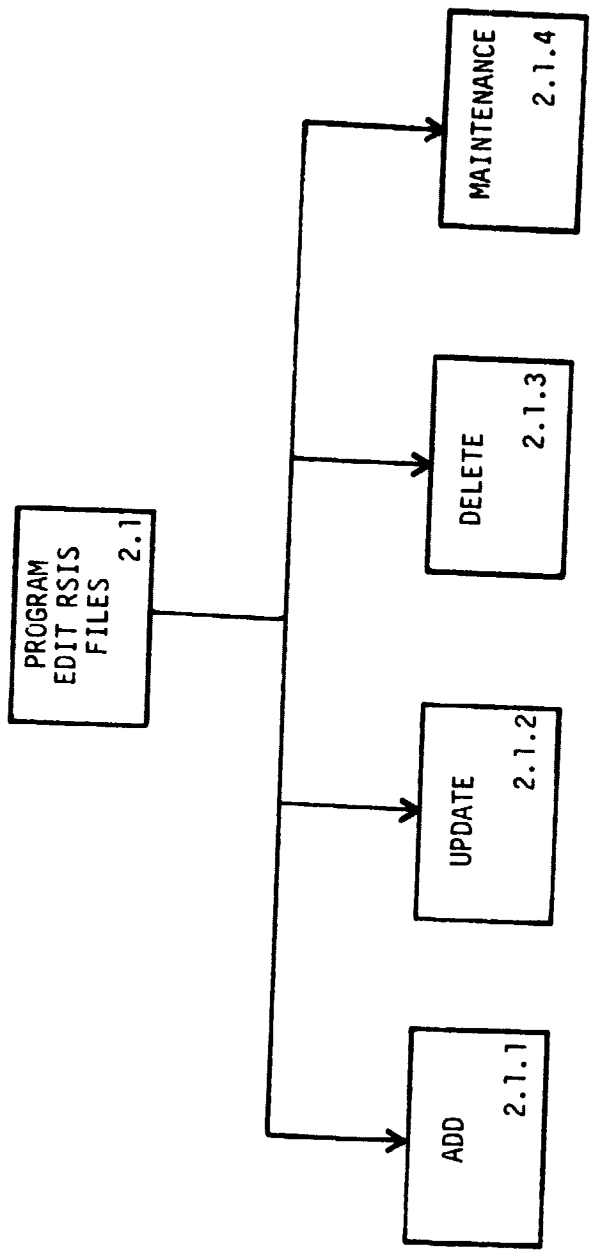

PROGRAM CHOICES
1.1.3
GENERATE LIST OF PROGRAMS NEEDED FOR OUTPUTS
DETERMINE AVAILABILITY OF REQUIRED INPUTS

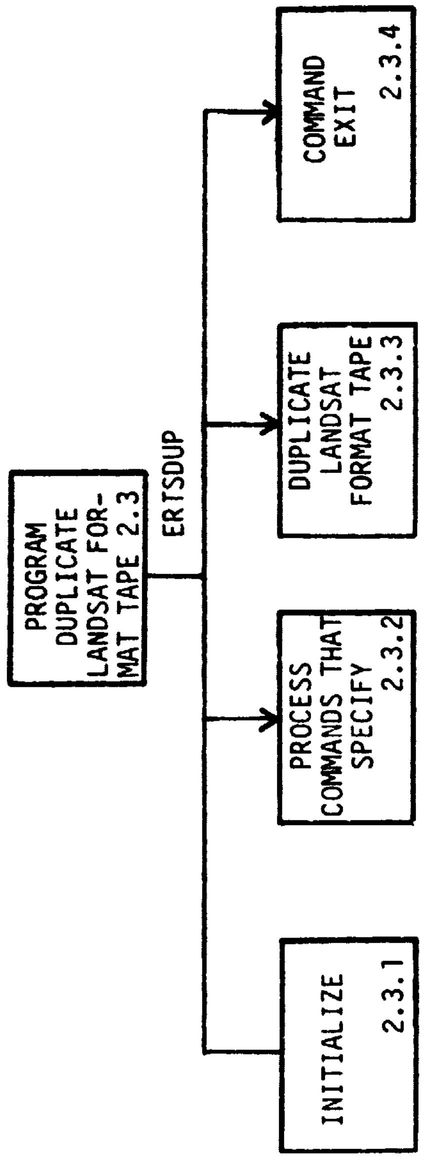

OUTPUT DIRECTIVES
1.1.4
VALIDATE RUN NUMBER
VALIDATE OUTPUT TITLES
VALIDATE OTHER RUN OPTIONS

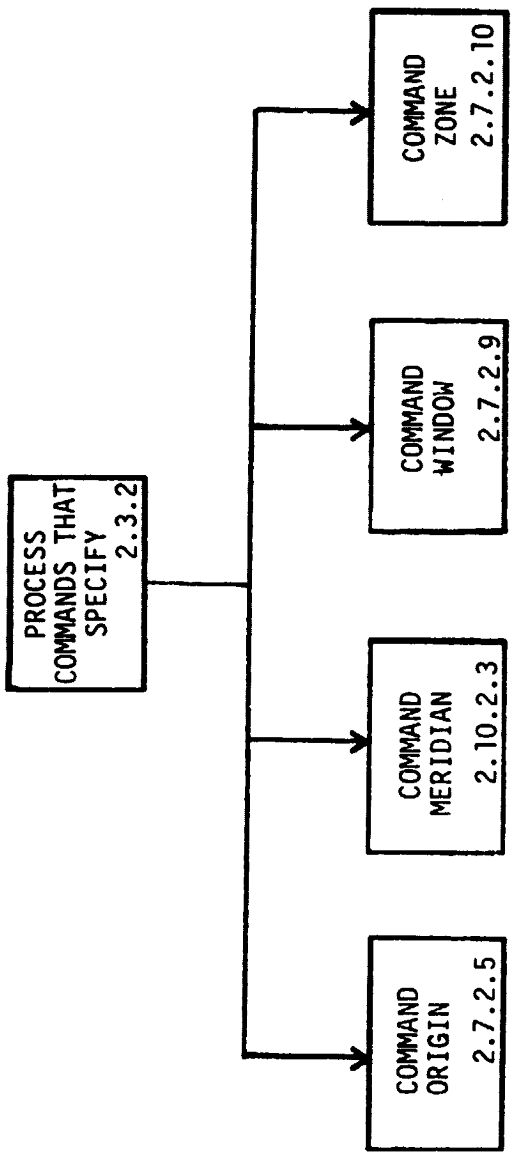

SET UP SYSTEM FILES
1.2
ASSIGN PROGRAM FILE
ASSIGN RUN INFORMATION FILE
ASSIGN DEFAULT FILE
ASSIGN RUN STREAM FILE

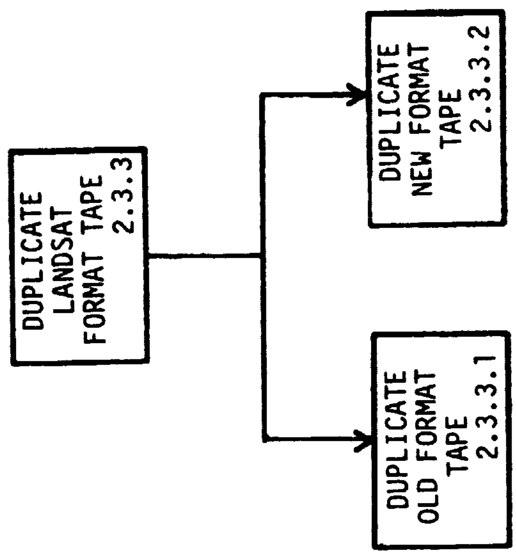

SET UP CONTROLS TO INITIALIZE REQUESTED PROGRAMS 1.4
VALIDATE RUN STREAM SYNTAX
MERGE INPUT SOURCES, DEFAULT FILE, RUN INFOR-MATION FILE
RENAME PROGRAM FILE
ADD ELEMENTS TO RUN STREAM FILE FROM RUN INFOR-MATION FILE

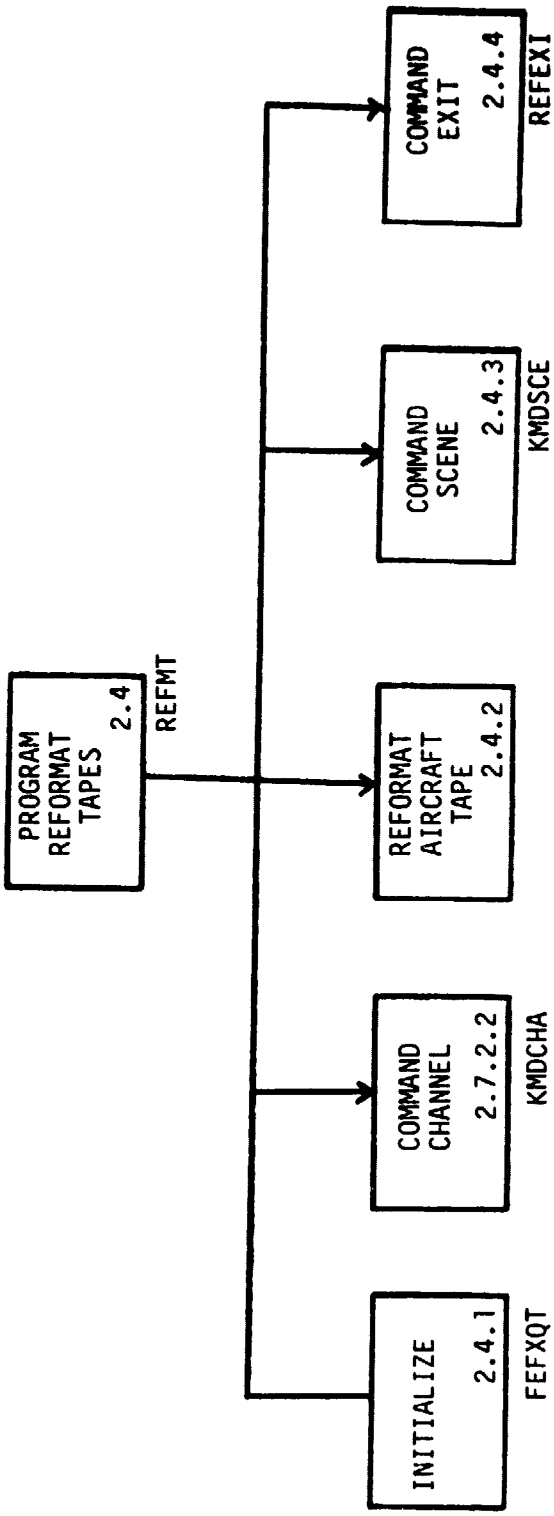

PROCESSING 2.0
EDIT FILES 2.1
PROGRAM SCENE ID 2.2
ERTSIDC
PROGRAM TAPE COPY 2.3
ERTSDUP
PROGRAM REFORMAT TAPES 2.4
REFMT
PROGRAM AD-JUST CONTROL NETWORK 2.5
CONTROL
PROGRAM PRINT DISPLAY AND PARTITION 2.6
PICTAB
PROGRAM COLOR DISPLAY AND PARTITION 2.7
CLRTAB
PROGRAM CLASSIFY DATA 2.8
CLASSIFY
PROGRAM PRINT DETECTION FILES 2.9
PRTDET
PROGRAM PRO-DUCE LINE PRINTER MAPS 2.10
PRTCLASS
PROGRAM CREATE PLOTS 2.11
PLTCLASS
PROGRAM CREATE FILM OUTPUT 2.12
FLMCLASS
PROGRAM PRO-DUCE COLOR CRT MAPS 2.13
CRTCLASS
PROGRAM ANOMALIES 2.14
ANOMLY
GRAPHIC FUNCTIONS 2.15

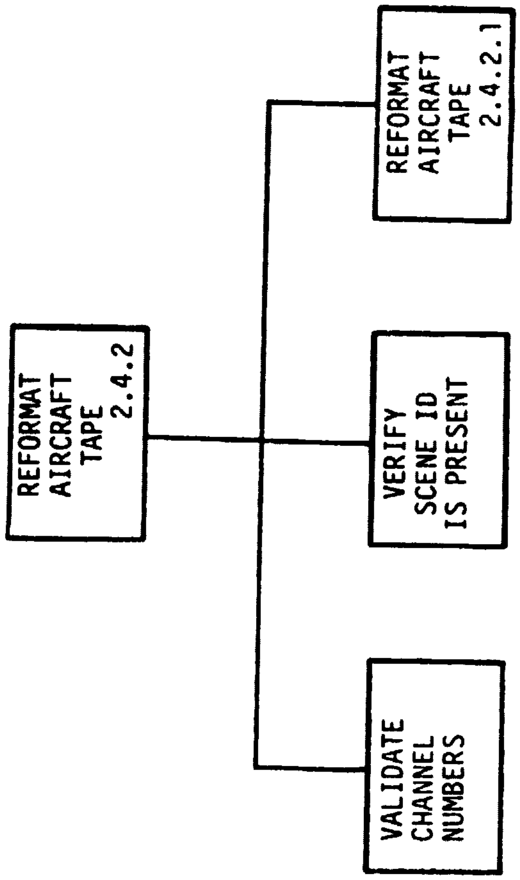

PROGRAM EDIT RSIS FILES 2.1
ADD 2.1.1
UPDATE 2.1.2
DELETE 2.1.3
MAINTENANCE 2.1.4

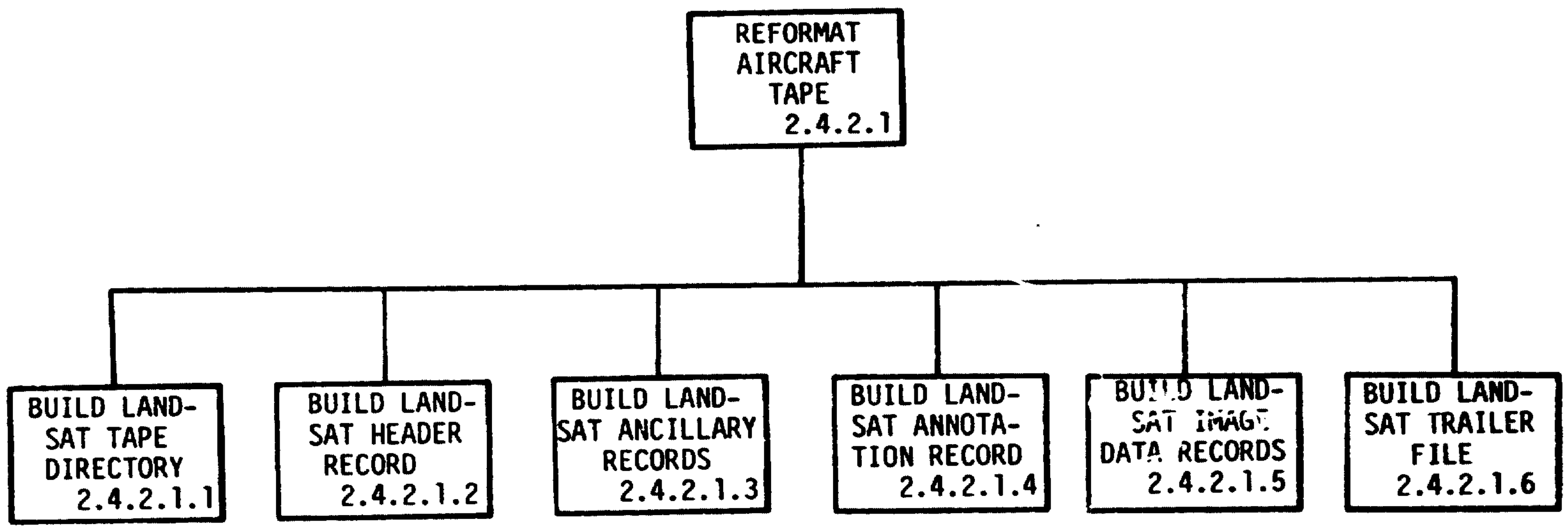

PROGRAM DUPLICATE LANDSAT FOR-MAT TAPE 2.3
ERTSDUP
INITIALIZE 2.3.1
PROCESS COMMANDS THAT SPECIFY 2.3.2
DUPLICATE LANDSAT FORMAT TAPE 2.3.3
COMMAND EXIT 2.3.4

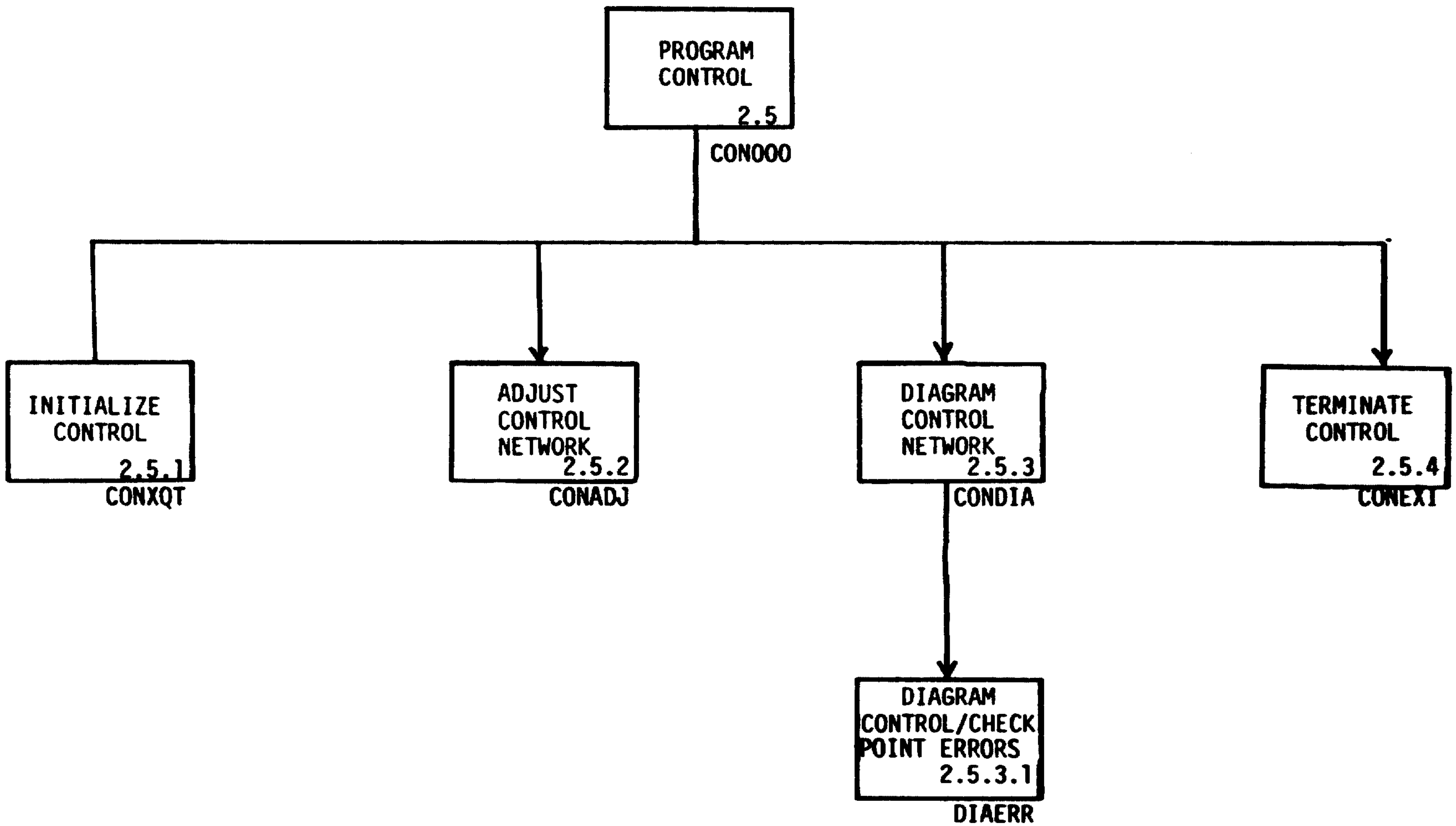

PROCESS COMMANDS THAT SPECIFY 2.3.2
COMMAND ORIGIN 2.7.2.5
COMMAND MERIDIAN 2.10.2.3
COMMAND WINDOW 2.7.2.9
COMMAND ZONE 2.7.2.10

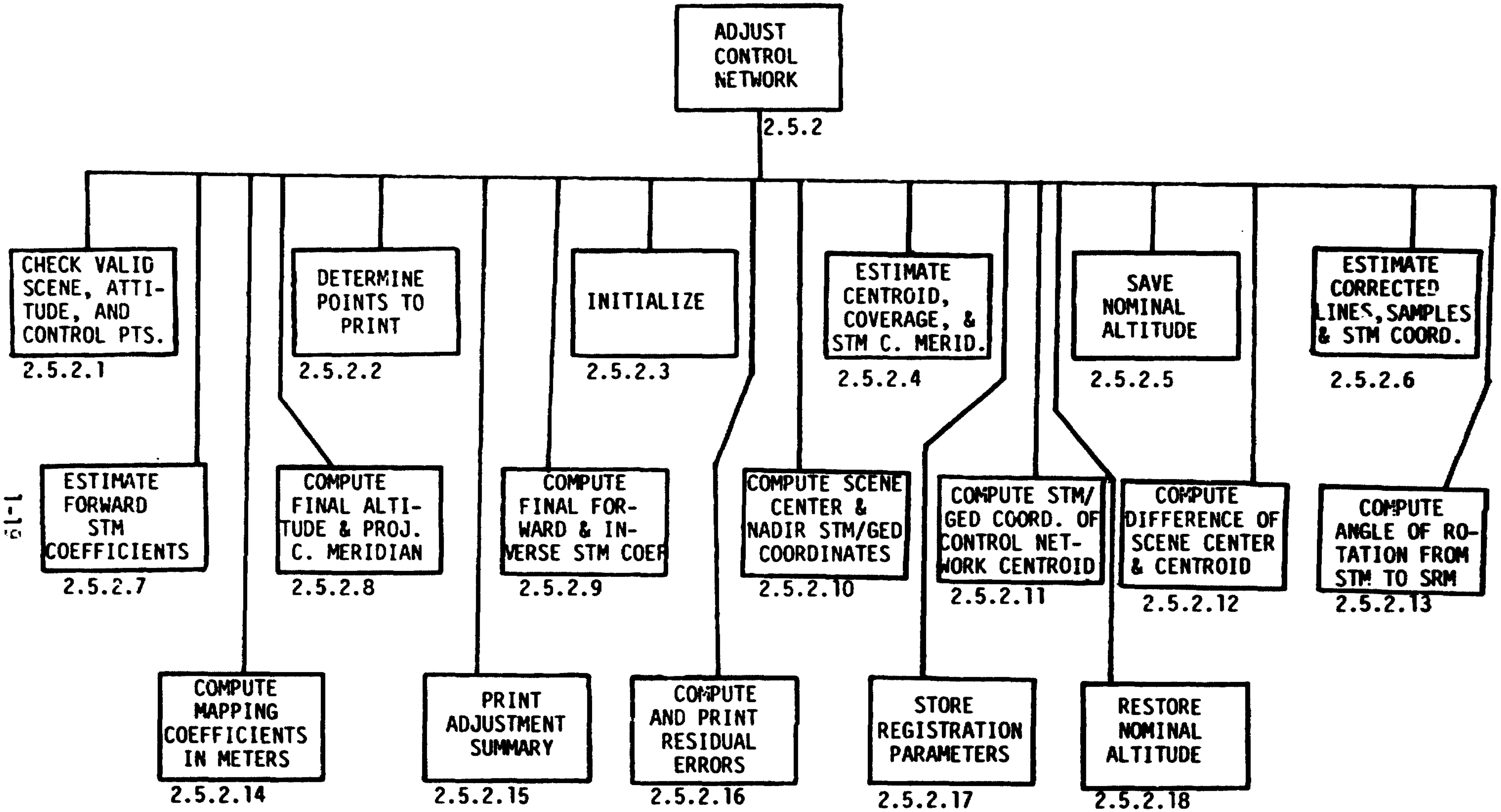

DUPLICATE LANDSAT FORMAT TAPE 2.3.3
DUPLICATE OLD FORMAT TAPE 2.3.3.1
DUPLICATE NEW FORMAT TAPE 2.3.3.2

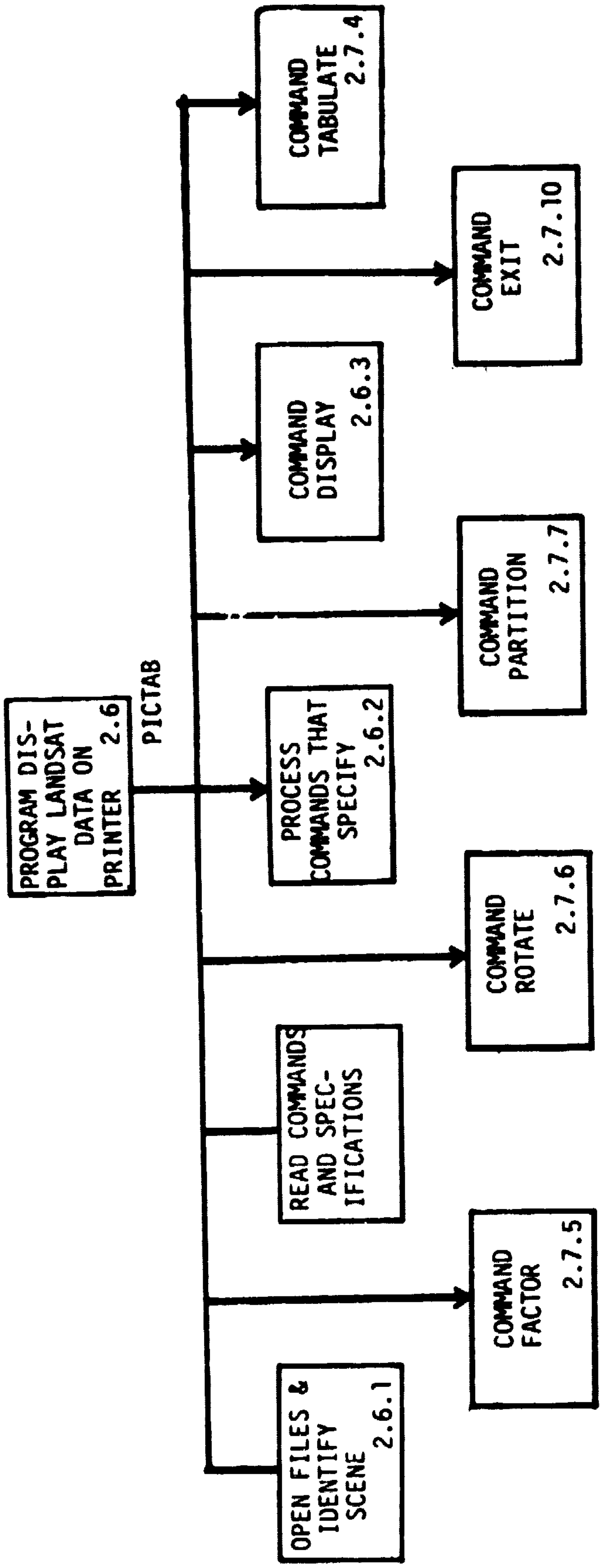

PROGRAM
REFORMAT
TAPES
2.4
REFMT
INITIALIZE
2.4.1
FEFXQT
COMMAND
CHANNEL
2.7.2.2
KMDCHA
REFORMAT
AIRCRAFT
TAPE
2.4.2
COMMAND
SCENE
2.4.3
KMDSCE
COMMAND
EXIT
2.4.4
REFEXI

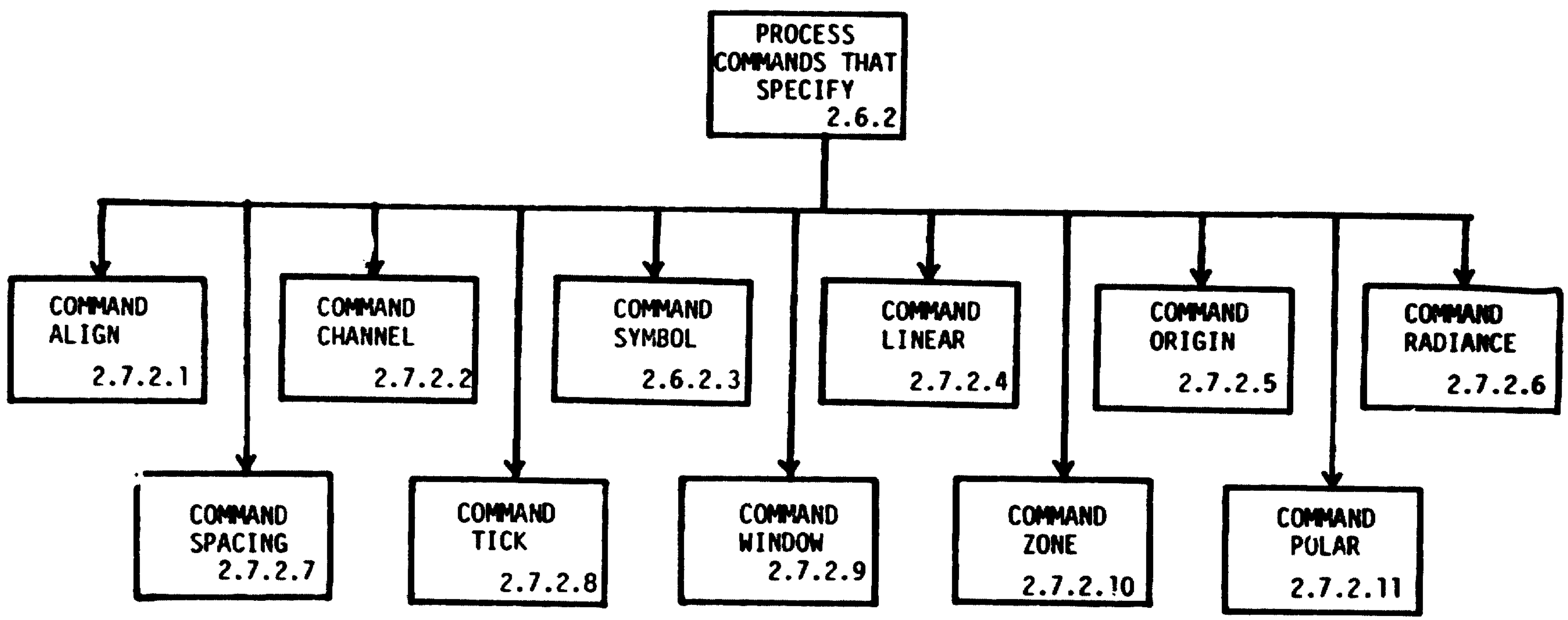

REFORMAT AIRCRAFT TAPE 2.4.2
REFORMAT AIRCRAFT TAPE 2.4.2.1
VERIFY SCENE ID IS PRESENT
VALIDATE CHANNEL NUMBERS

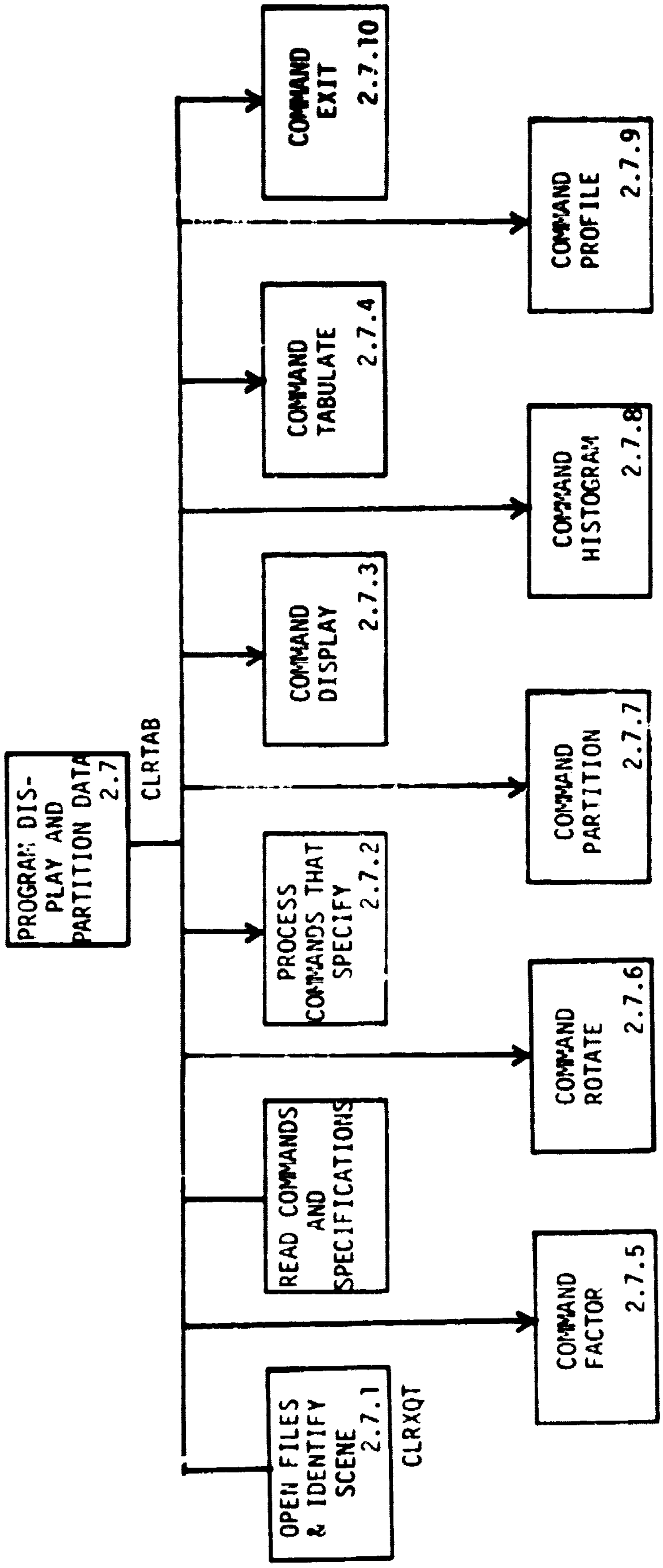

REFORMAT AIRCRAFT TAPE 2.4.2.1
BUILD LAND-SAT TAPE DIRECTORY 2.4.2.1.1
BUILD LAND-SAT HEADER RECORD 2.4.2.1.2
BUILD LAND-SAT ANCILLARY RECORDS 2.4.2.1.3
BUILD LAND-SAT ANNOTATION RECORD 2.4.2.1.4
BUILD LAND-SAT IMAGE DATA RECORDS 2.4.2.1.5
BUILD LAND-SAT TRAILER FILE 2.4.2.1.6

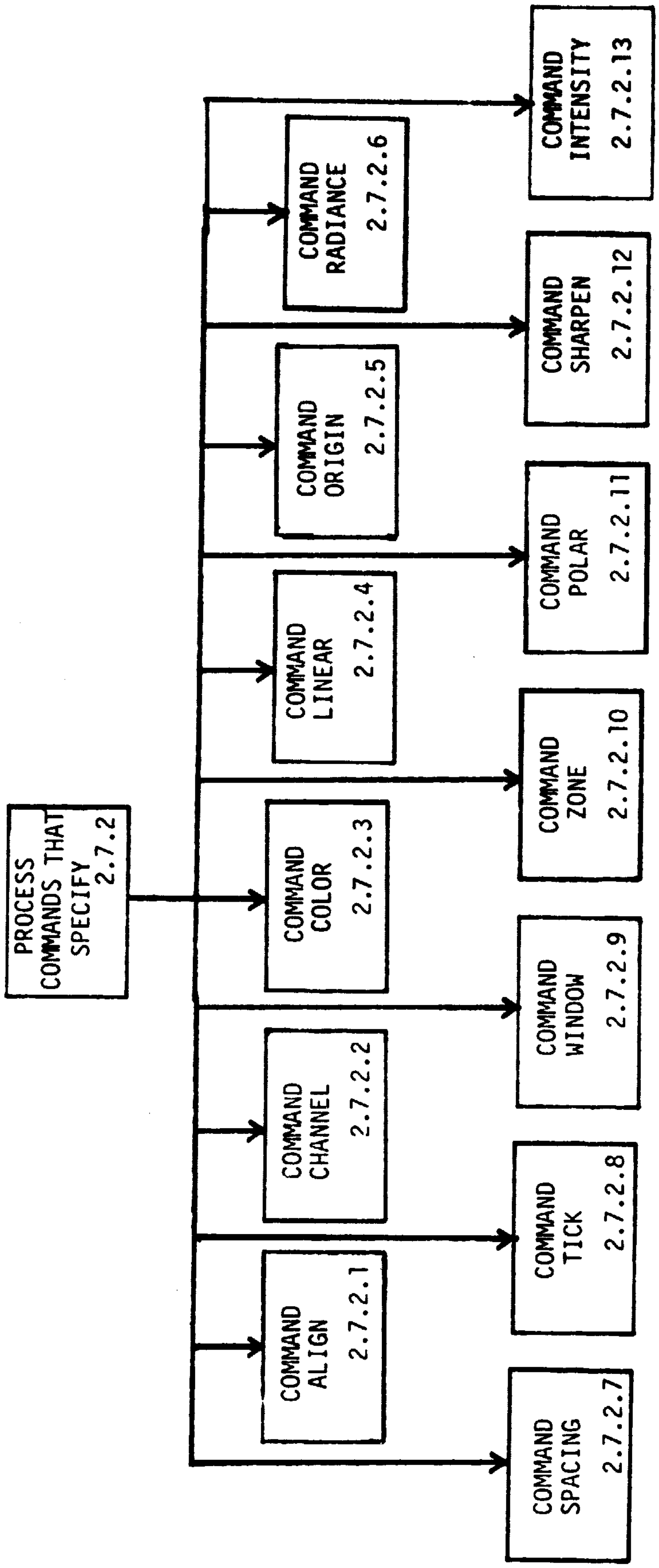

PROGRAM CONTROL
2.5
CONOOO
INITIALIZE CONTROL
2.5.1
CONXQT
ADJUST CONTROL NETWORK
2.5.2
CONADJ
DIAGRAM CONTROL NETWORK
2.5.3
CONDIA
TERMINATE CONTROL
2.5.4
CONEXI
DIAGRAM CONTROL/CHECK POINT ERRORS
2.5.3.1
DIAERR

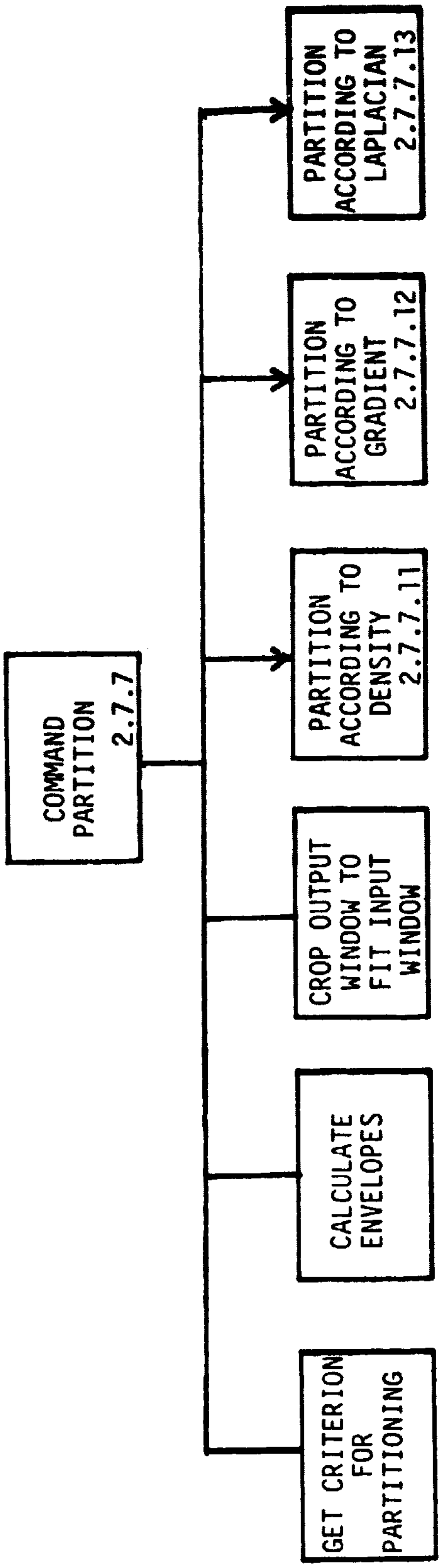

ADJUST CONTROL NETWORK
2.5.2
CHECK VALID SCENE, ATTITUDE, AND CONTROL PTS.
2.5.2.1
DETERMINE POINTS TO PRINT
2.5.2.2
INITIALIZE
2.5.2.3
ESTIMATE CENTROID, COVERAGE, & STM C. MERID.
2.5.2.4
SAVE NOMINAL ALTITUDE
2.5.2.5
ESTIMATE CORRECTED LINES, SAMPLES & STM COORD.
2.5.2.6
ESTIMATE FORWARD STM COEFFICIENTS
2.5.2.7
COMPUTE FINAL ALTITUDE & PROJ. C. MERIDIAN
2.5.2.8
COMPUTE FINAL FORWARD & INVERSE STM COEF
2.5.2.9
COMPUTE SCENE CENTER & NADIR STM/GED COORDINATES
2.5.2.10
COMPUTE STM/GED COORD. OF CONTROL NETWORK CENTROID
2.5.2.11
COMPUTE DIFFERENCE OF SCENE CENTER & CENTROID
2.5.2.12
COMPUTE ANGLE OF ROTATION FROM STM TO SRM
2.5.2.13
COMPUTE MAPPING COEFFICIENTS IN METERS
2.5.2.14
PRINT ADJUSTMENT SUMMARY
2.5.2.15
COMPUTE AND PRINT RESIDUAL ERRORS
2.5.2.16
STORE REGISTRATION PARAMETERS
2.5.2.17
RESTORE NOMINAL ALTITUDE
2.5.2.18

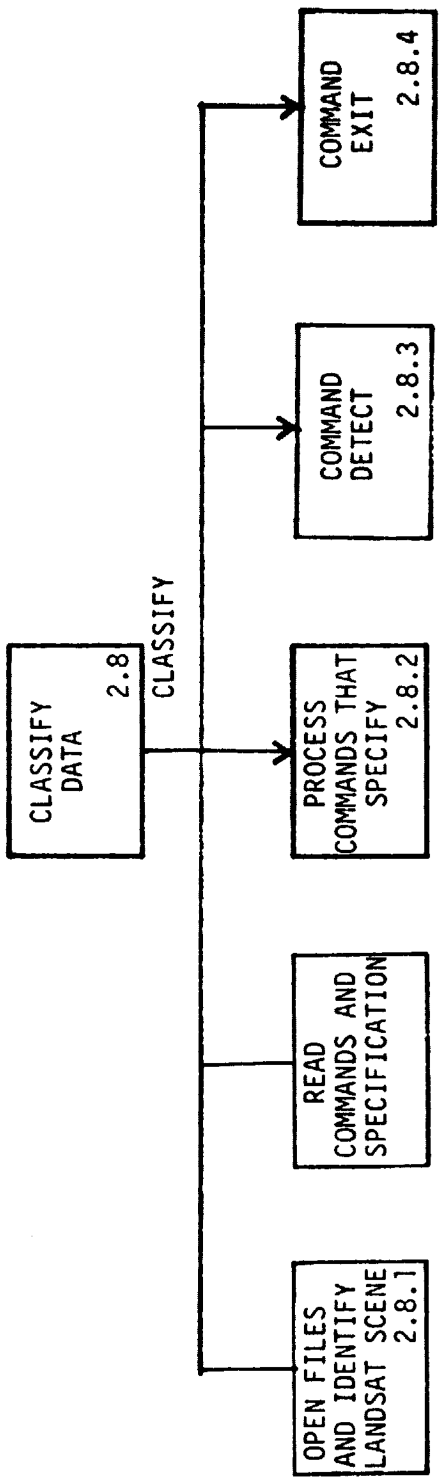

PICTAB
PROGRAM DIS-PLAY LANDSAT DATA ON PRINTER 2.6
OPEN FILES & IDENTIFY SCENE 2.6.1
READ COMMANDS AND SPEC-IFICATIONS
PROCESS COMMANDS THAT SPECIFY 2.6.2
COMMAND DISPLAY 2.6.3
COMMAND TABULATE 2.7.4
COMMAND FACTOR 2.7.5
COMMAND ROTATE 2.7.6
COMMAND PARTITION 2.7.7
COMMAND EXIT 2.7.10

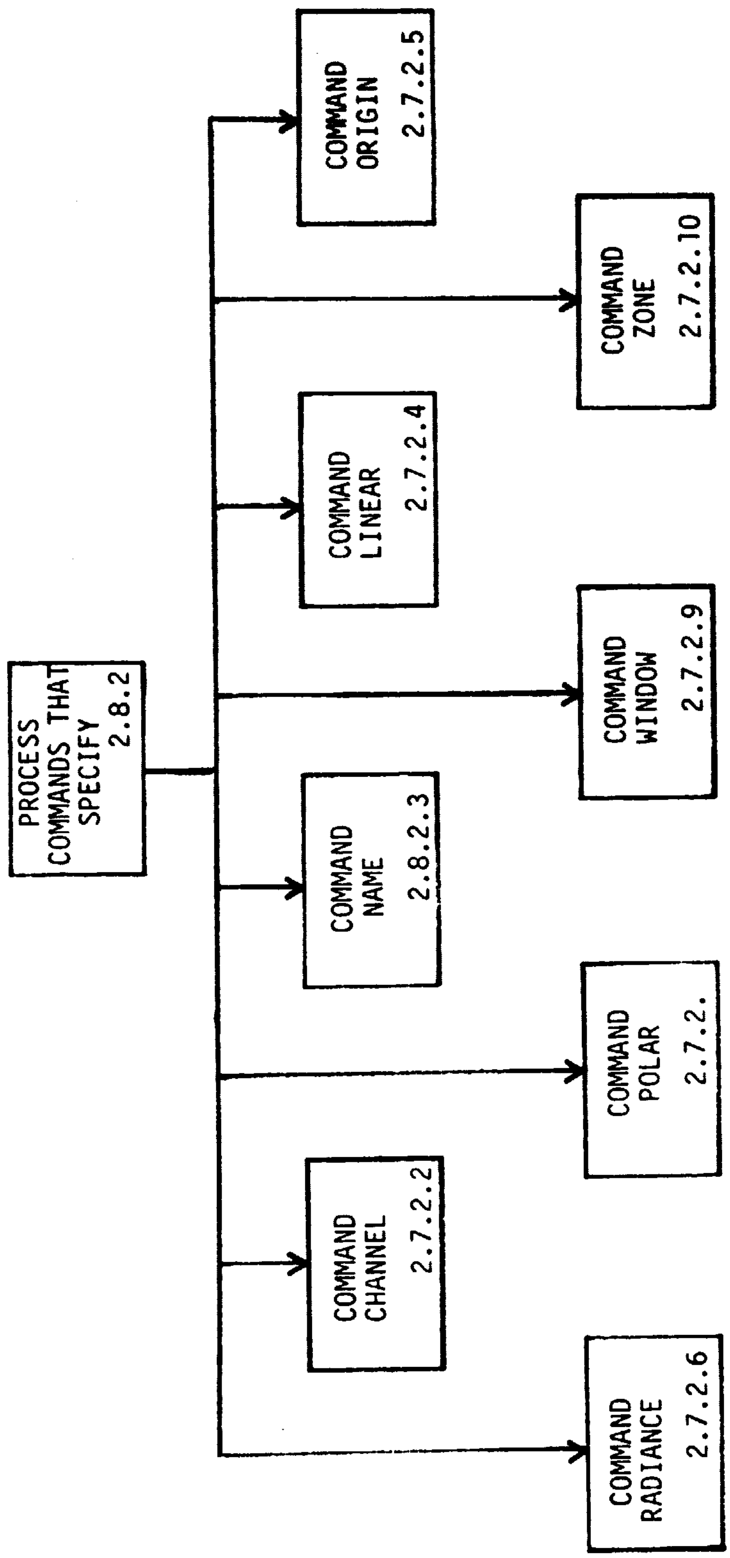

PROCESS COMMANDS THAT SPECIFY 2.6.2
COMMAND ALIGN 2.7.2.1
COMMAND SPACING 2.7.2.7
COMMAND CHANNEL 2.7.2.2
COMMAND TICK 2.7.2.8
COMMAND SYMBOL 2.6.2.3
COMMAND WINDOW 2.7.2.9
COMMAND LINEAR 2.7.2.4
COMMAND ZONE 2.7.2.10
COMMAND ORIGIN 2.7.2.5
COMMAND POLAR 2.7.2.11
COMMAND RADIANCE 2.7.2.6

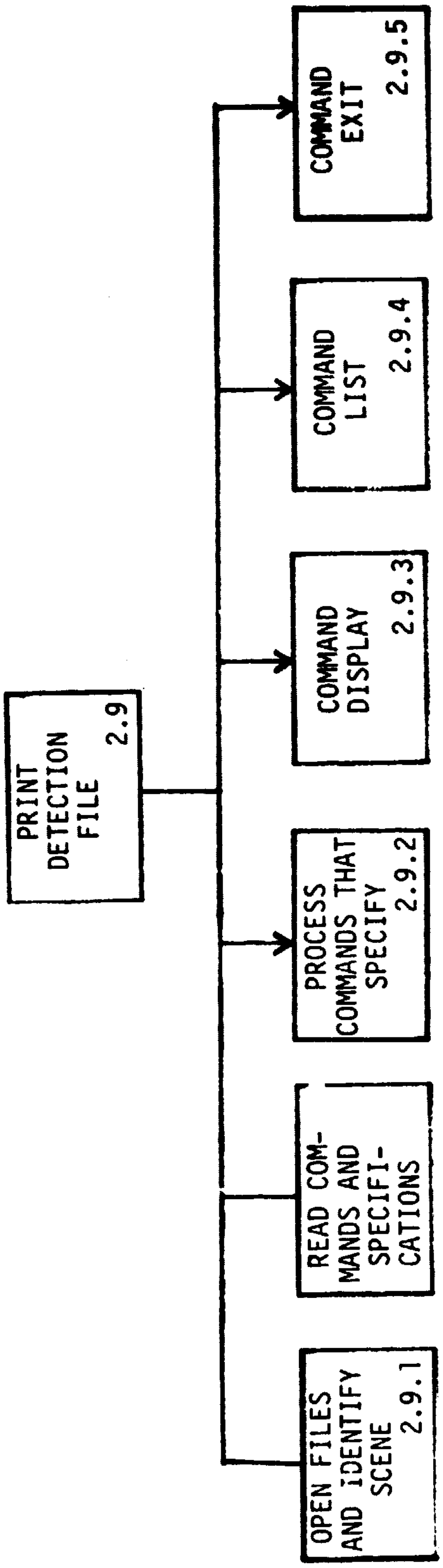

PROGRAM: DISPLAY AND PARTITION DATA 2.7
CLRTAB
OPEN FILES & IDENTIFY SCENE 2.7.1
CLRXQT
READ COMMANDS AND SPECIFICATIONS
PROCESS COMMANDS THAT SPECIFY 2.7.2
COMMAND DISPLAY 2.7.3
COMMAND TABULATE 2.7.4
COMMAND EXIT 2.7.10
COMMAND FACTOR 2.7.5
COMMAND ROTATE 2.7.6
COMMAND PARTITION 2.7.7
COMMAND HISTOGRAM 2.7.8
COMMAND PROFILE 2.7.9

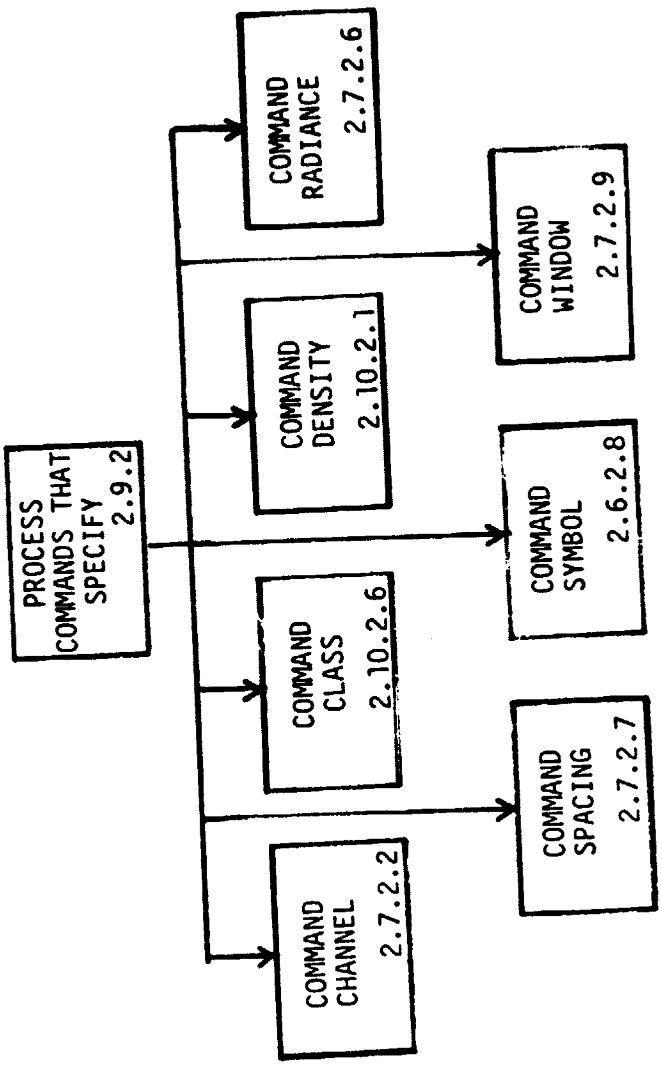

PROCESS COMMANDS THAT SPECIFY 2.7.2
COMMAND ALIGN 2.7.2.1
COMMAND CHANNEL 2.7.2.2
COMMAND COLOR 2.7.2.3
COMMAND LINEAR 2.7.2.4
COMMAND ORIGIN 2.7.2.5
COMMAND RADIANCE 2.7.2.6
COMMAND SPACING 2.7.2.7
COMMAND TICK 2.7.2.8
COMMAND WINDOW 2.7.2.9
COMMAND ZONE 2.7.2.10
COMMAND POLAR 2.7.2.11
COMMAND SHARPEN 2.7.2.12
COMMAND INTENSITY 2.7.2.13

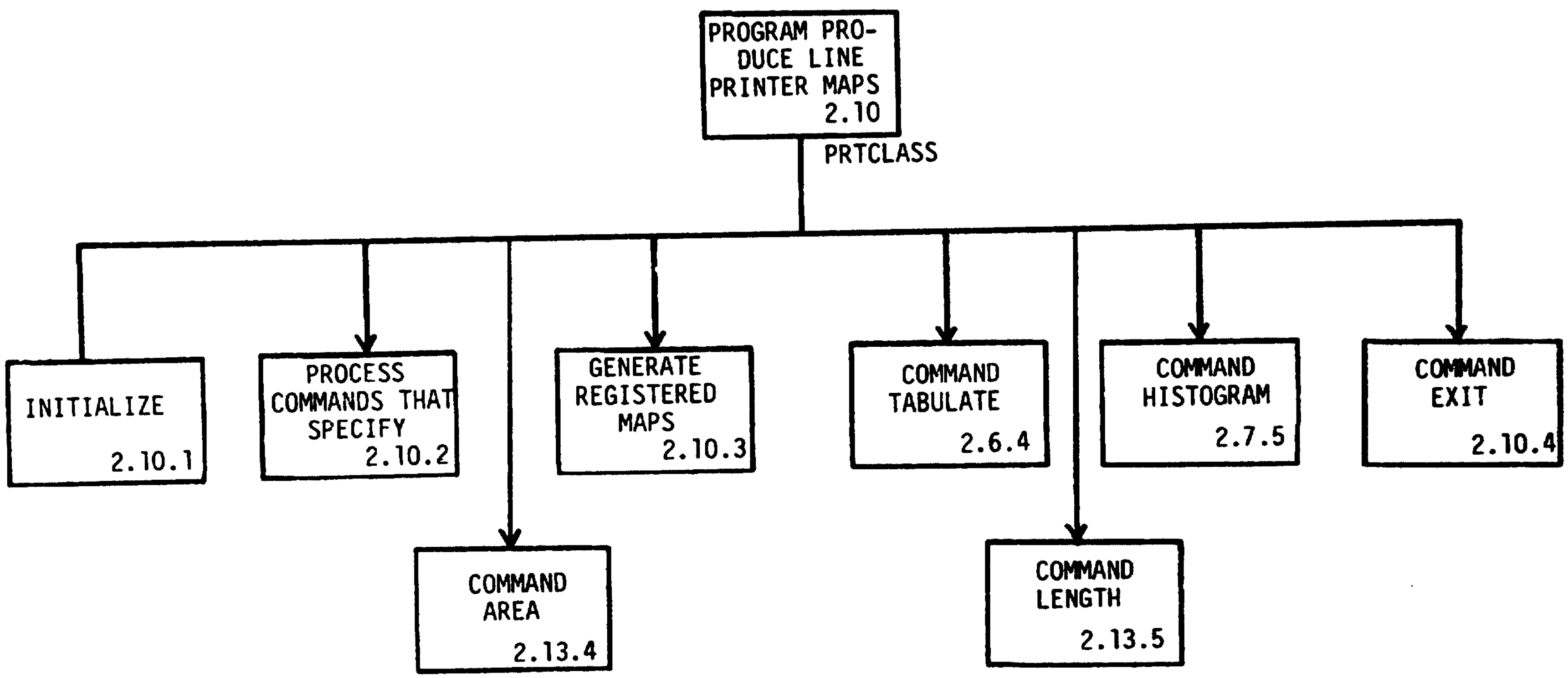

COMMAND PARTITION 2.7.7
PARTITION ACCORDING TO DENSITY 2.7.7.11
PARTITION ACCORDING TO GRADIENT 2.7.7.12
PARTITION ACCORDING TO LAPLACIAN 2.7.7.13
CROP OUTPUT WINDOW TO FIT INPUT WINDOW
CALCULATE ENVELOPES
GET CRITERION FOR PARTITIONING

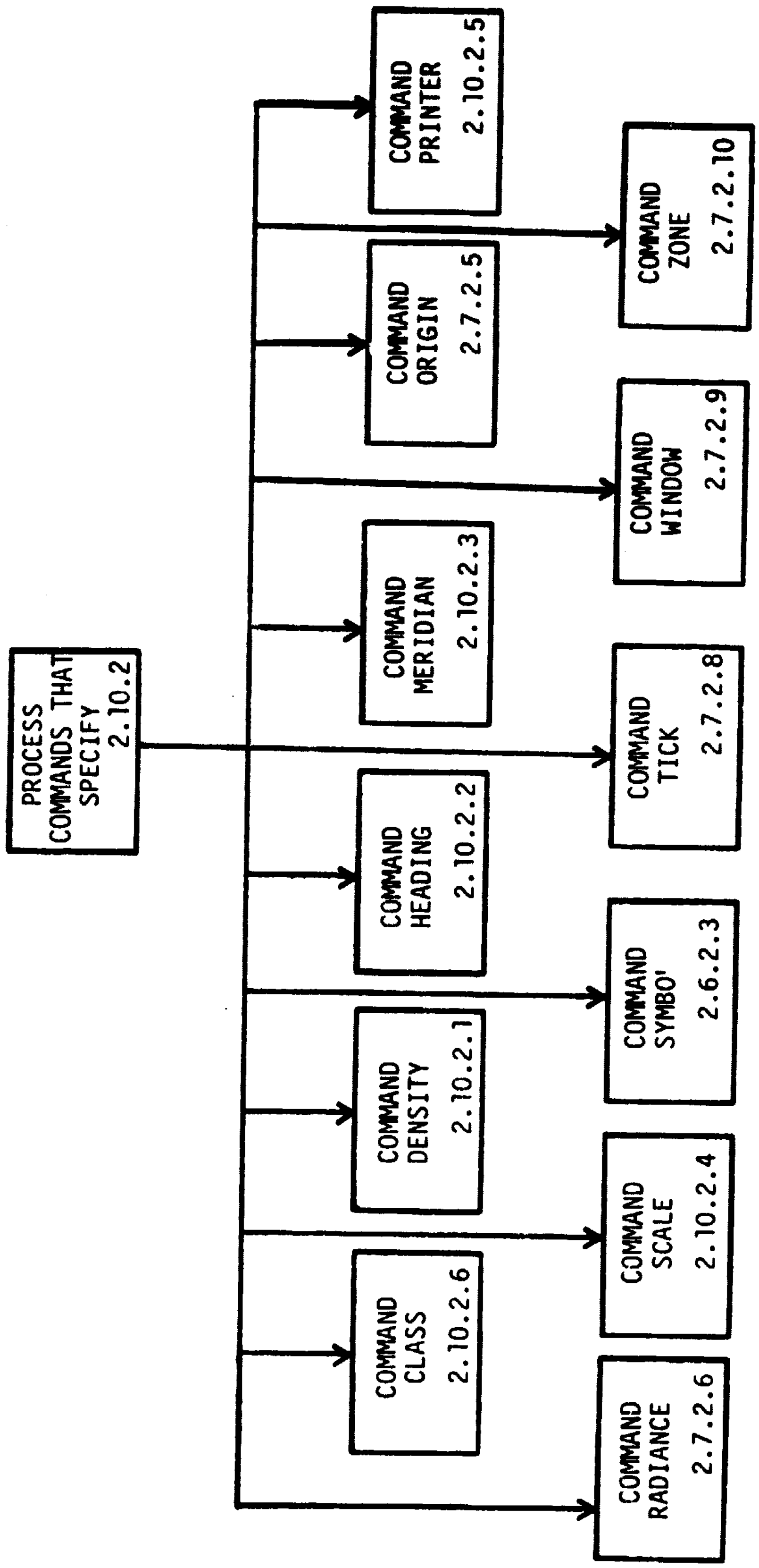

CLASSIFY DATA 2.8
CLASSIFY
OPEN FILES AND IDENTIFY LANDSAT SCENE 2.8.1
READ COMMANDS AND SPECIFICATION
PROCESS COMMANDS THAT SPECIFY 2.8.2
COMMAND DETECT 2.8.3
COMMAND EXIT 2.8.4

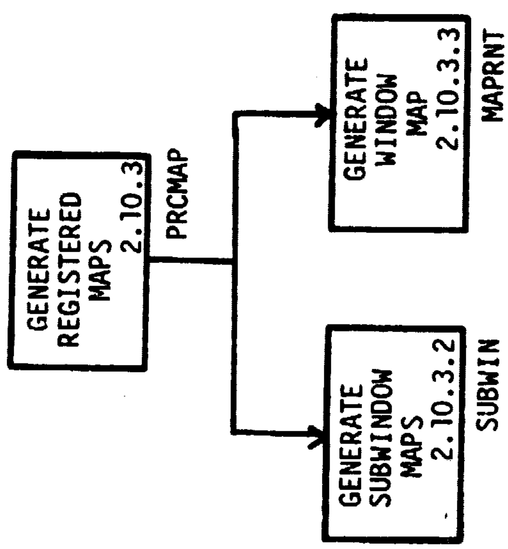

PROCESS COMMANDS THAT SPECIFY
2.8.2
COMMAND ORIGIN
2.7.2.5
COMMAND ZONE
2.7.2.10
COMMAND LINEAR
2.7.2.4
COMMAND WINDOW
2.7.2.9
COMMAND NAME
2.8.2.3
COMMAND POLAR
2.7.2.
COMMAND CHANNEL
2.7.2.2
COMMAND RADIANCE
2.7.2.6

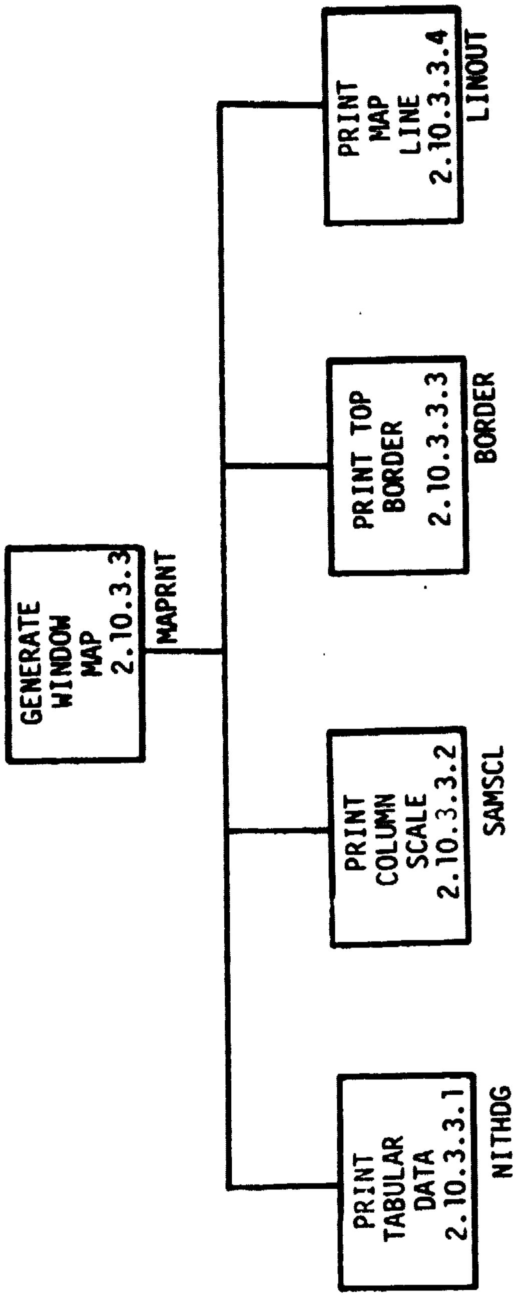

PRINT DETECTION FILE 2.9
OPEN FILES AND IDENTIFY SCENE 2.9.1
READ COMMANDS AND SPECIFICATIONS
PROCESS COMMANDS THAT SPECIFY 2.9.2
COMMAND DISPLAY 2.9.3
COMMAND LIST 2.9.4
COMMAND EXIT 2.9.5

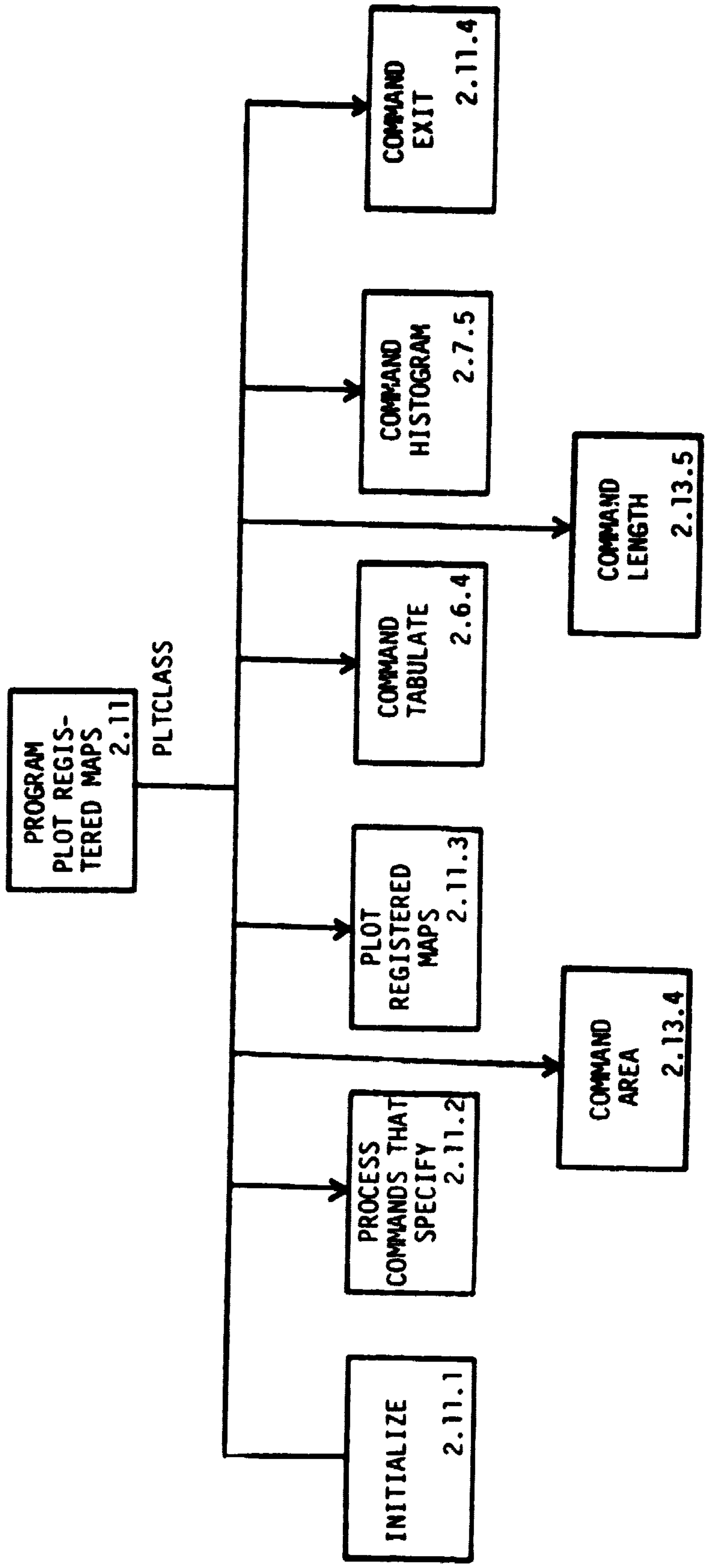

PROCESS COMMANDS THAT SPECIFY 2.9.2
COMMAND RADIANCE 2.7.2.6
COMMAND WINDOW 2.7.2.9
COMMAND DENSITY 2.10.2.1
COMMAND SYMBOL 2.6.2.8
COMMAND CLASS 2.10.2.6
COMMAND SPACING 2.7.2.7
COMMAND CHANNEL 2.7.2.2

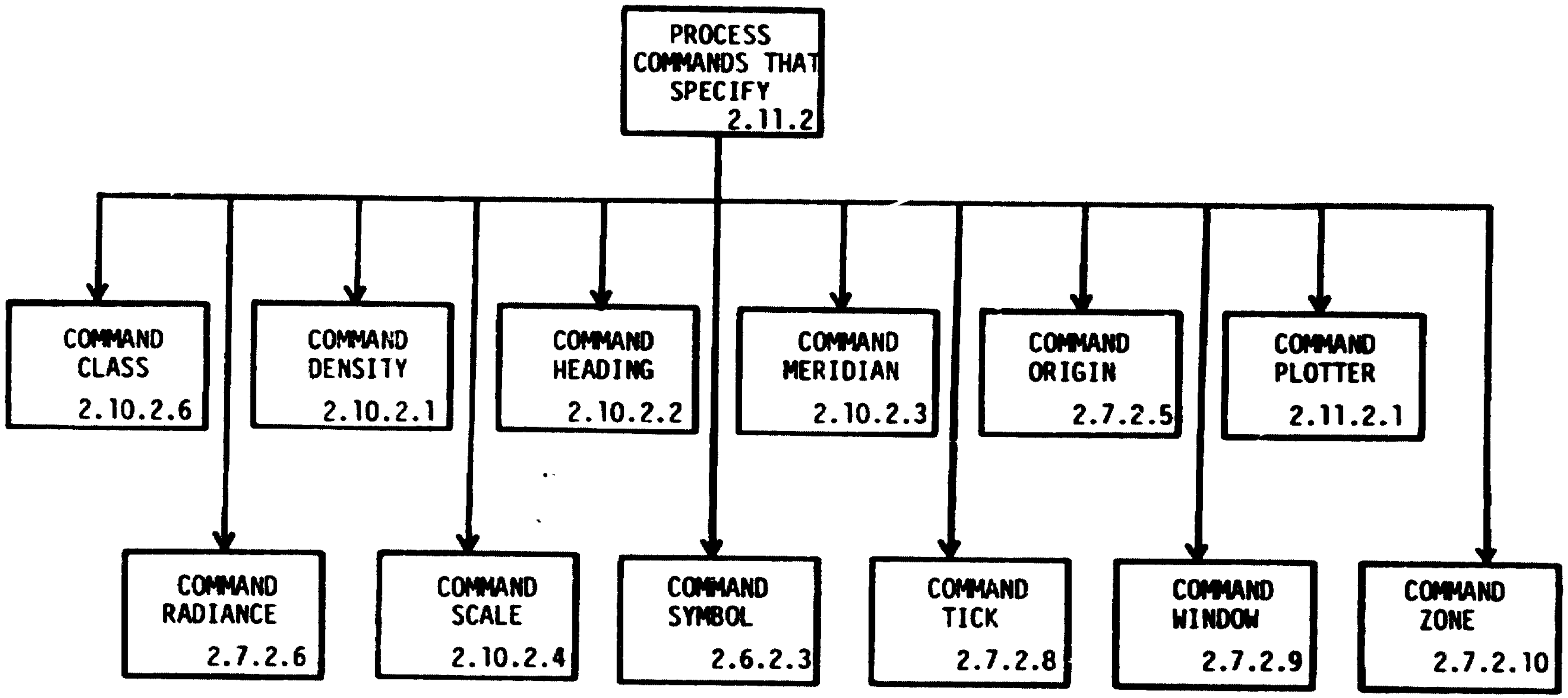

PROGRAM PRODUCE LINE PRINTER MAPS 2.10
PRTCLASS
INITIALIZE 2.10.1
PROCESS COMMANDS THAT SPECIFY 2.10.2
COMMAND AREA 2.13.4
GENERATE REGISTERED MAPS 2.10.3
COMMAND TABULATE 2.6.4
COMMAND LENGTH 2.13.5
COMMAND HISTOGRAM 2.7.5
COMMAND EXIT 2.10.4

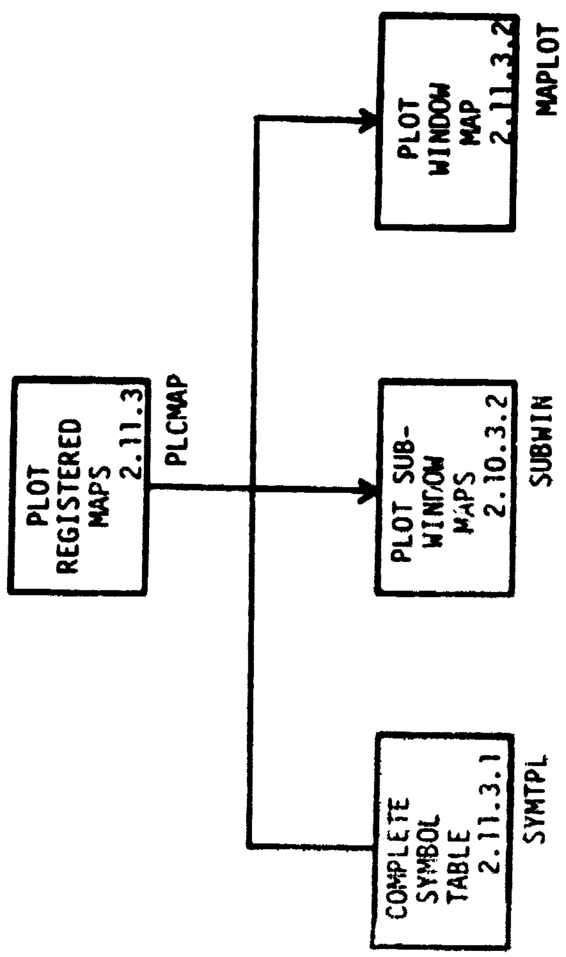

PROCESS COMMANDS THAT SPECIFY 2.10.2
COMMAND CLASS 2.10.2.6
COMMAND DENSITY 2.10.2.1
COMMAND HEADING 2.10.2.2
COMMAND MERIDIAN 2.10.2.3
COMMAND ORIGIN 2.7.2.5
COMMAND PRINTER 2.10.2.5
COMMAND RADIANCE 2.7.2.6
COMMAND SCALE 2.10.2.4
COMMAND SYMBO' 2.6.2.3
COMMAND TICK 2.7.2.8
COMMAND WINDOW 2.7.2.9
COMMAND ZONE 2.7.2.10

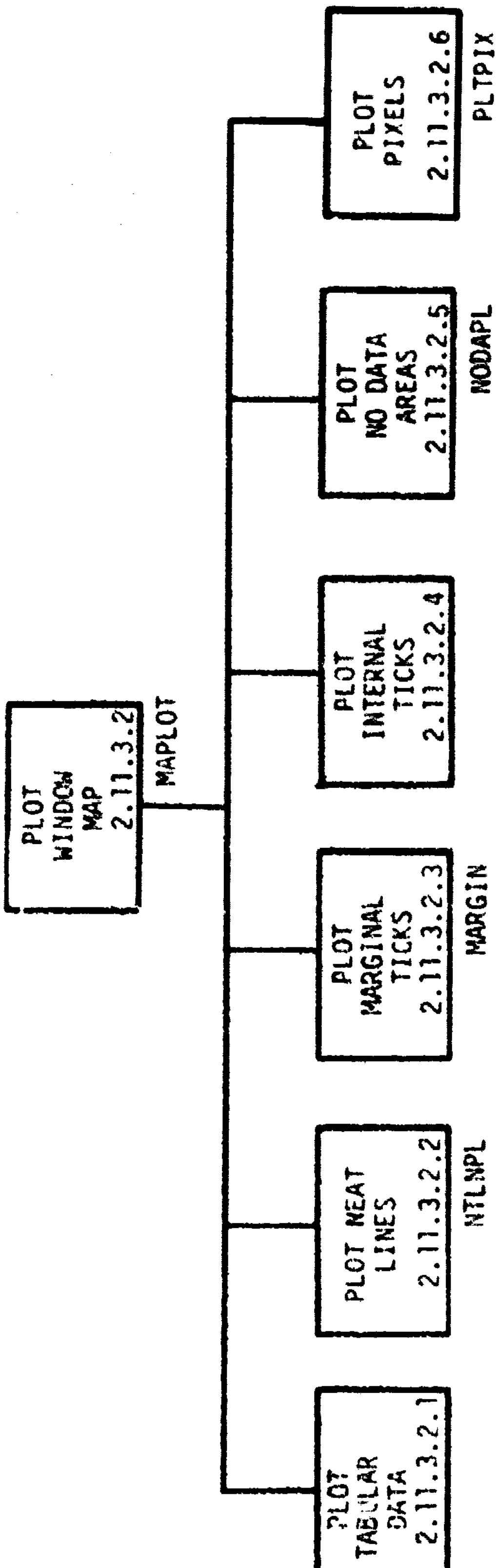

GENERATE REGISTERED MAPS 2.10.3
PRCMAP
GENERATE SUBWINDOW MAPS 2.10.3.2
SUBWIN
GENERATE WINDOW MAP 2.10.3.3
MAPRNT

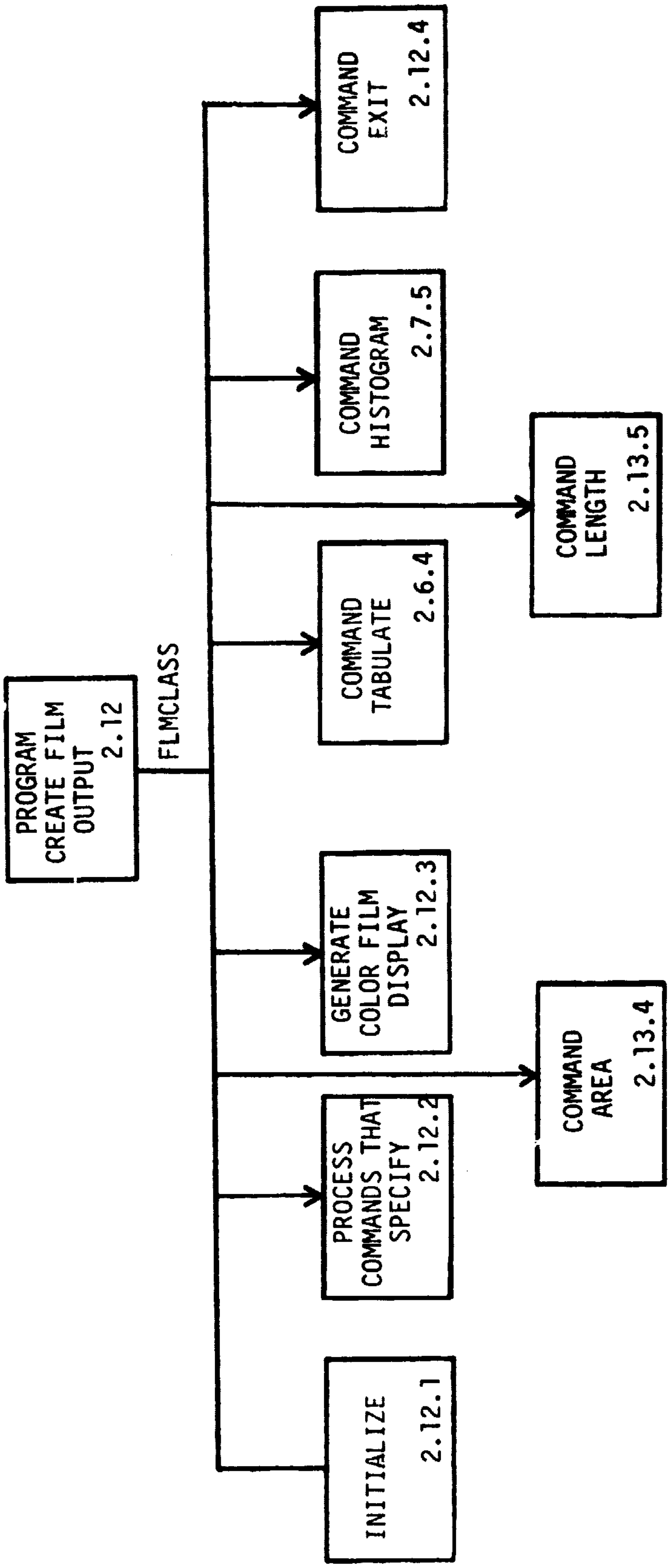

GENERATE WINDOW MAP 2.10.3.3
MAPRNT
PRINT TABULAR DATA 2.10.3.3.1
NITHDG
PRINT COLUMN SCALE 2.10.3.3.2
SAMSCL
PRINT TOP BORDER 2.10.3.3.3
BORDER
PRINT MAP LINE 2.10.3.3.4
LINOUT

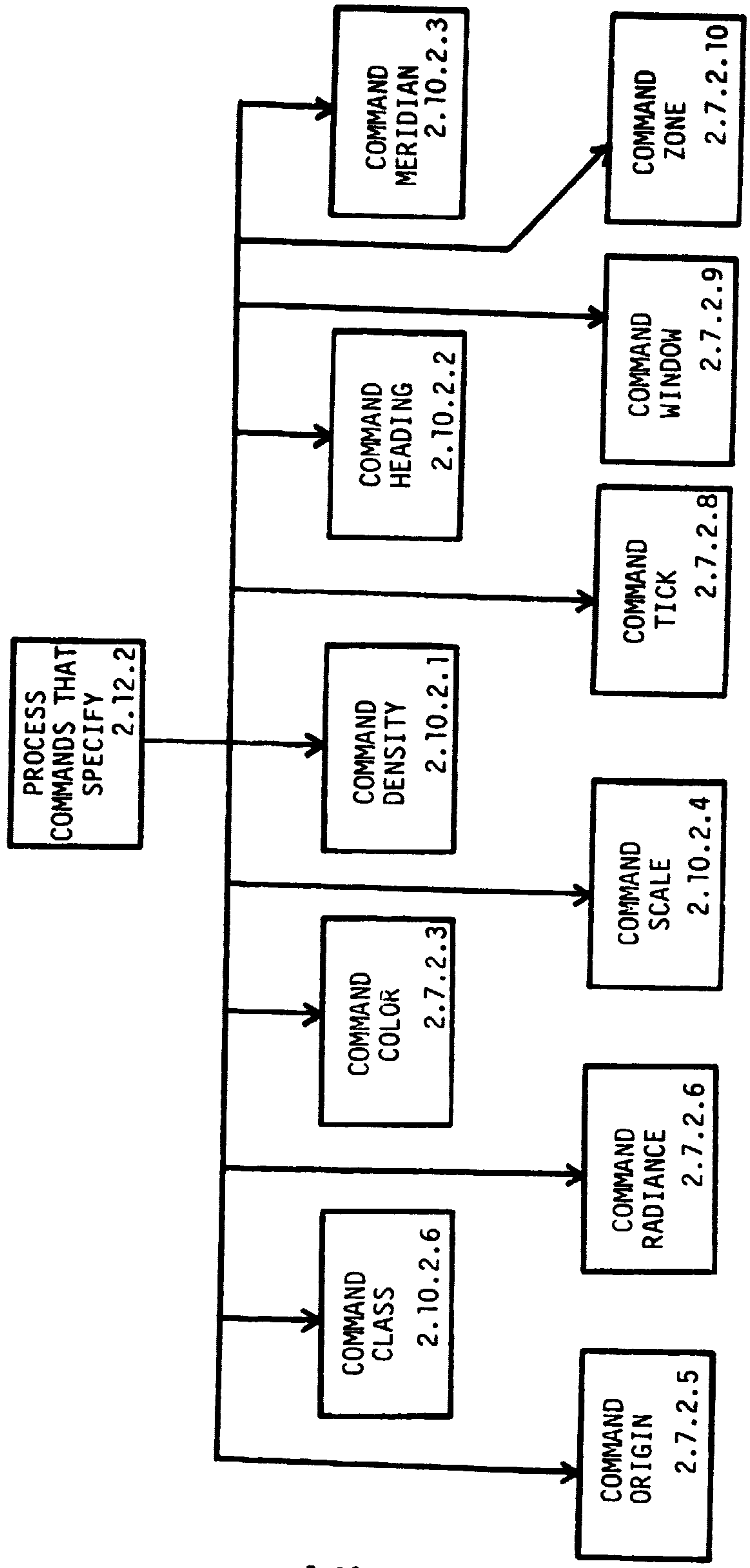

PROGRAM PLOT REGIS- TERED MAPS 2.11
PLTCLASS
INITIALIZE 2.11.1
PROCESS COMMANDS THAT SPECIFY 2.11.2
COMMAND AREA 2.13.4
PLOT REGISTERED MAPS 2.11.3
COMMAND TABULATE 2.6.4
COMMAND LENGTH 2.13.5
COMMAND HISTOGRAM 2.7.5
COMMAND EXIT 2.11.4

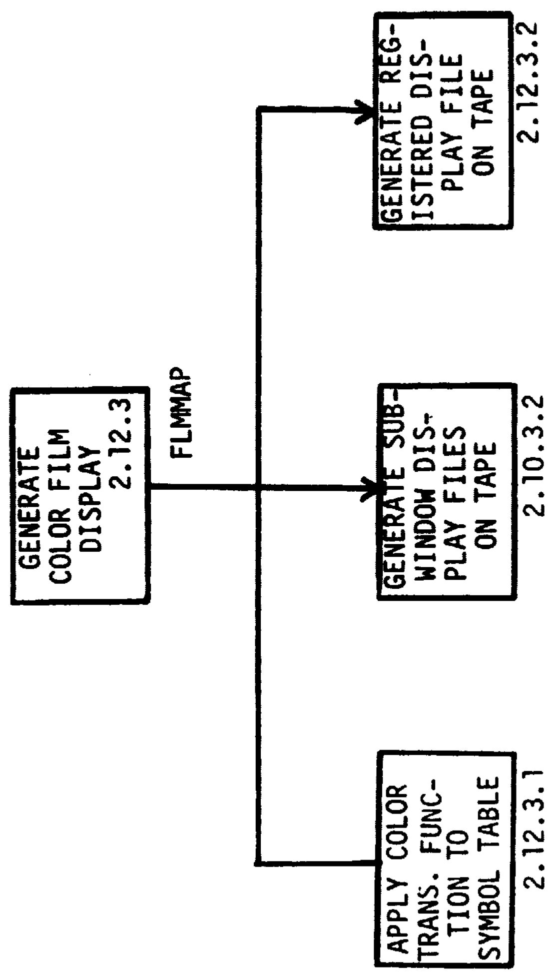

PROCESS COMMANDS THAT SPECIFY 2.11.2
COMMAND CLASS 2.10.2.6
COMMAND RADIANCE 2.7.2.6
COMMAND DENSITY 2.10.2.1
COMMAND SCALE 2.10.2.4
COMMAND HEADING 2.10.2.2
COMMAND SYMBOL 2.6.2.3
COMMAND MERIDIAN 2.10.2.3
COMMAND TICK 2.7.2.8
COMMAND ORIGIN 2.7.2.5
COMMAND WINDOW 2.7.2.9
COMMAND PLOTTER 2.11.2.1
COMMAND ZONE 2.7.2.10

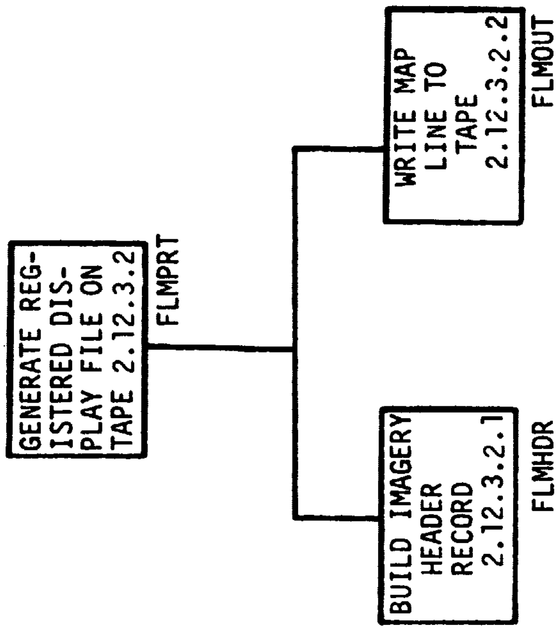

PLOT REGISTERED MAPS 2.11.3
PLCMAP
COMPLETE SYMBOL TABLE 2.11.3.1
SYMTPL
PLOT SUB-WINDOW MAPS 2.10.3.2
SUBWIN
PLOT WINDOW MAP 2.11.3.2
MAPLOT

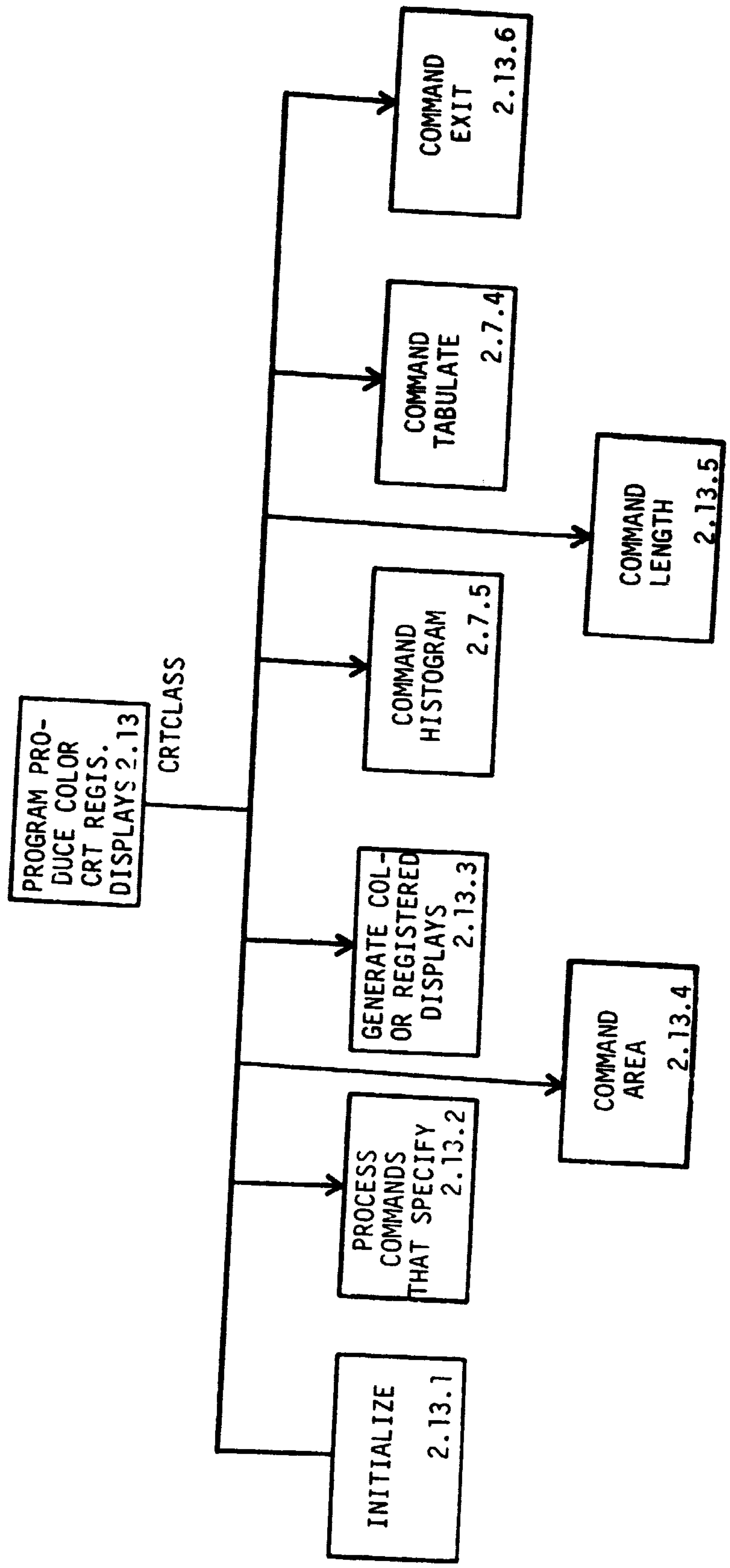

PLOT WINDOW MAP 2.11.3.2
MAPLOT
PLOT TABULAR DATA 2.11.3.2.1
PLOT NEAT LINES 2.11.3.2.2
NTLNPL
PLOT MARGINAL TICKS 2.11.3.2.3
MARGIN
PLOT INTERNAL TICKS 2.11.3.2.4
PLOT NO DATA AREAS 2.11.3.2.5
NODAPL
PLOT PIXELS 2.11.3.2.6
PLTPIX

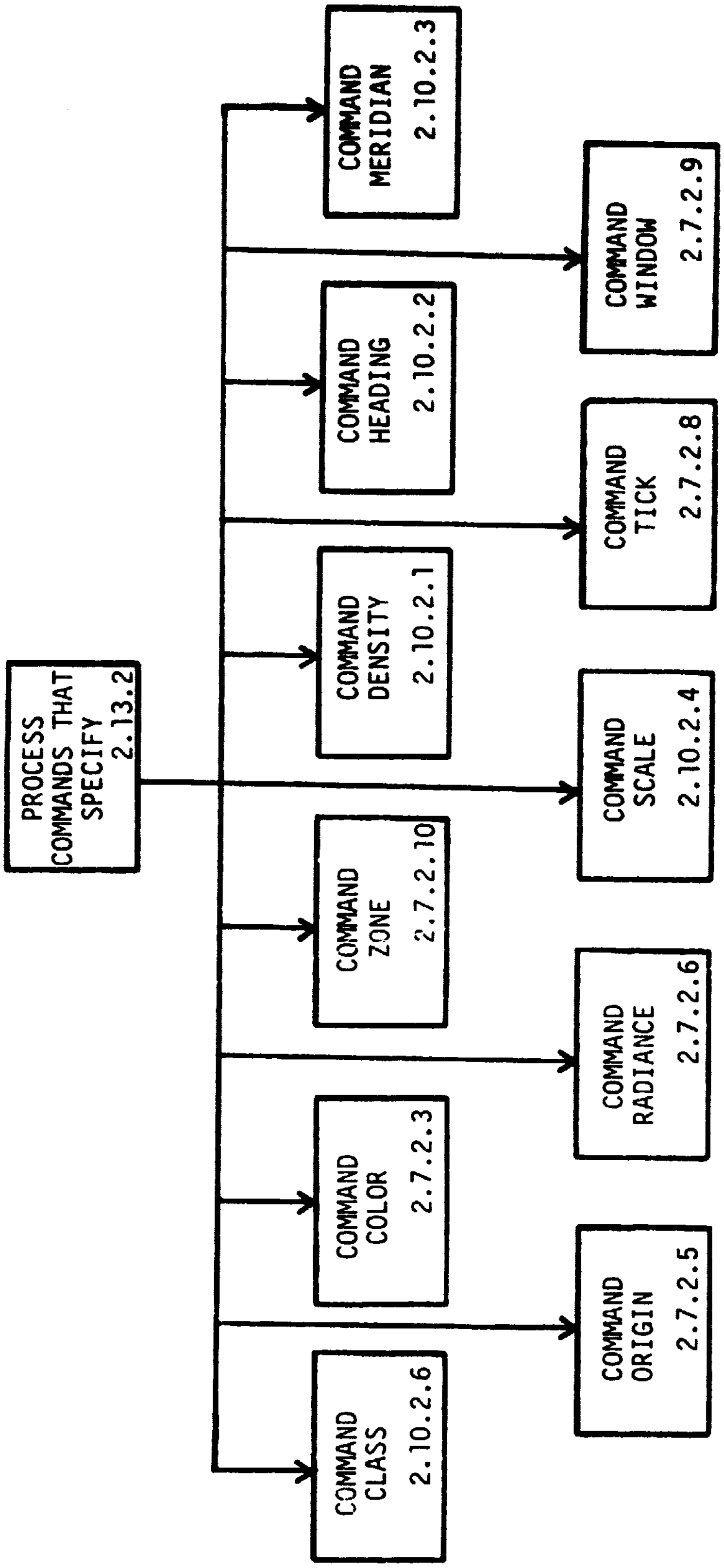

PROGRAM
CREATE FILM
OUTPUT
2.12
FLMCLASS
INITIALIZE
2.12.1
PROCESS
COMMANDS THAT
SPECIFY
2.12.2
COMMAND
AREA
2.13.4
GENERATE
COLOR FILM
DISPLAY
2.12.3
COMMAND
TABULATE
2.6.4
COMMAND
LENGTH
2.13.5
COMMAND
HISTOGRAM
2.7.5
COMMAND
EXIT
2.12.4

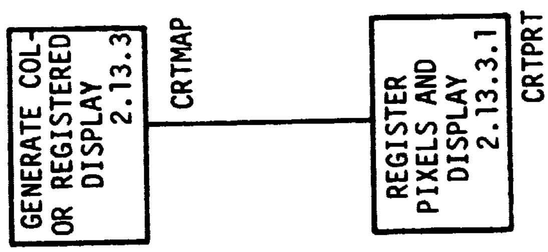

PROCESS COMMANDS THAT SPECIFY 2.12.2
COMMAND MERIDIAN 2.10.2.3
COMMAND ZONE 2.7.2.10
COMMAND WINDOW 2.7.2.9
COMMAND HEADING 2.10.2.2
COMMAND TICK 2.7.2.8
COMMAND DENSITY 2.10.2.1
COMMAND SCALE 2.10.2.4
COMMAND COLOR 2.7.2.3
COMMAND RADIANCE 2.7.2.6
COMMAND CLASS 2.10.2.6
COMMAND ORIGIN 2.7.2.5

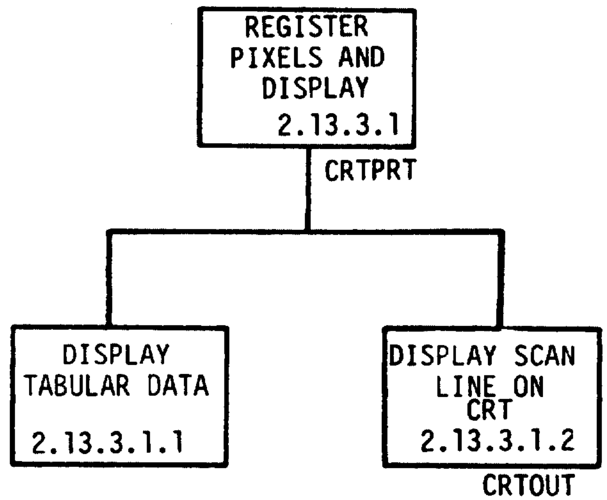

GENERATE COLOR FILM DISPLAY 2.12.3
FLMMAP
APPLY COLOR TRANS. FUNC- TION TO SYMBOL TABLE 2.12.3.1
GENERATE SUB- WINDOW DIS- PLAY FILES ON TAPE 2.10.3.2
GENERATE REG- ISTERED DIS- PLAY FILE ON TAPE 2.12.3.2

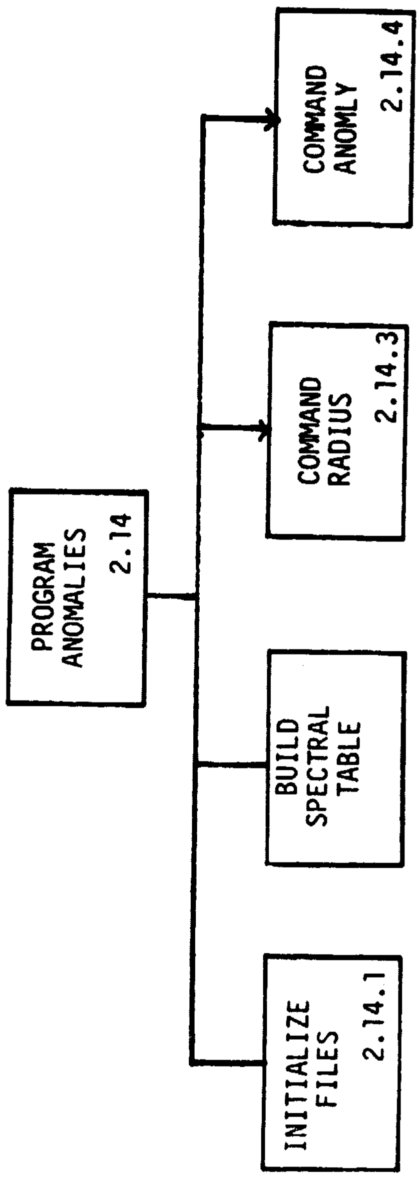

GENERATE REG-ISTERED DIS-PLAY FILE ON TAPE 2.12.3.2
FLMPRT
BUILD IMAGERY HEADER RECORD 2.12.3.2.1
FLMHDR
WRITE MAP LINE TO TAPE 2.12.3.2.2
FLMOUT

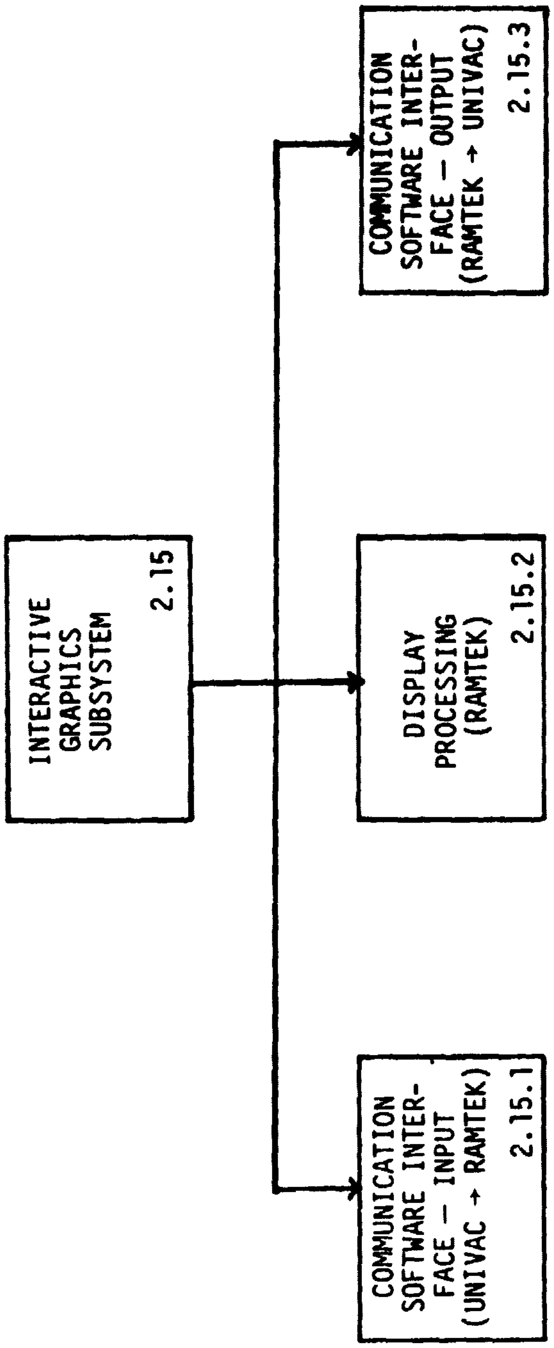

PROGRAM PRODUCE COLOR CRT REGIS. DISPLAYS 2.13
CRTCLASS
INITIALIZE 2.13.1
PROCESS COMMANDS THAT SPECIFY 2.13.2
COMMAND AREA 2.13.4
GENERATE COLOR REGISTERED DISPLAYS 2.13.3
COMMAND HISTOGRAM 2.7.5
COMMAND LENGTH 2.13.5
COMMAND TABULATE 2.7.4
COMMAND EXIT 2.13.6

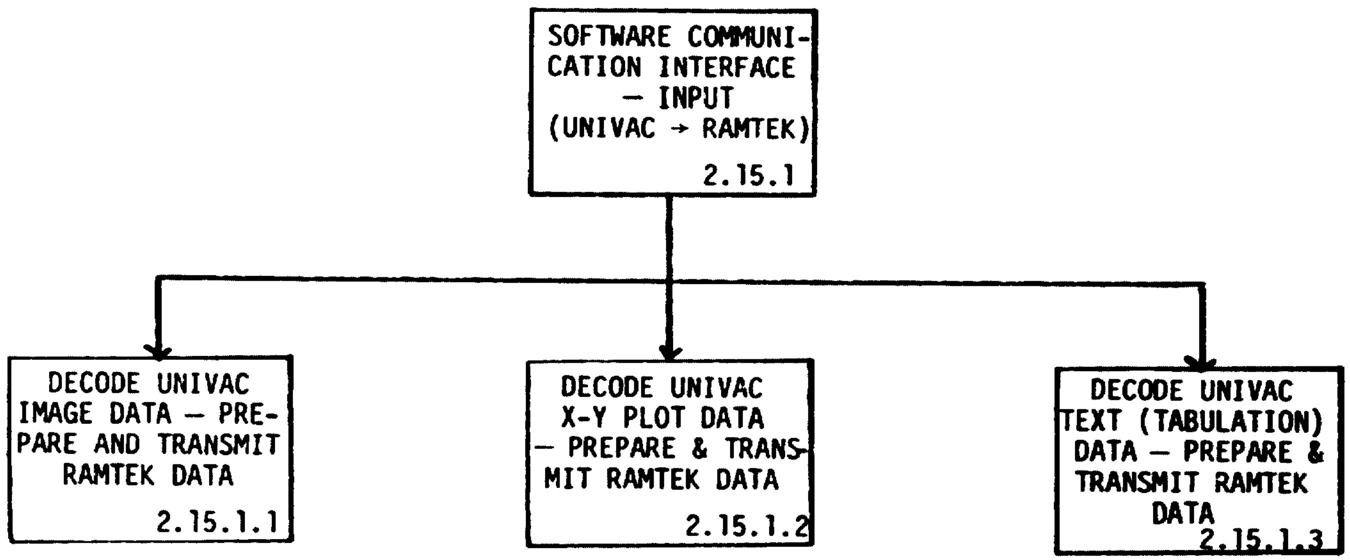

PROCESS COMMANDS THAT SPECIFY 2.13.2
COMMAND CLASS 2.10.2.6
COMMAND ORIGIN 2.7.2.5
COMMAND COLOR 2.7.2.3
COMMAND RADIANCE 2.7.2.6
COMMAND ZONE 2.7.2.10
COMMAND SCALE 2.10.2.4
COMMAND DENSITY 2.10.2.1
COMMAND TICK 2.7.2.8
COMMAND HEADING 2.10.2.2
COMMAND WINDOW 2.7.2.9
COMMAND MERIDIAN 2.10.2.3

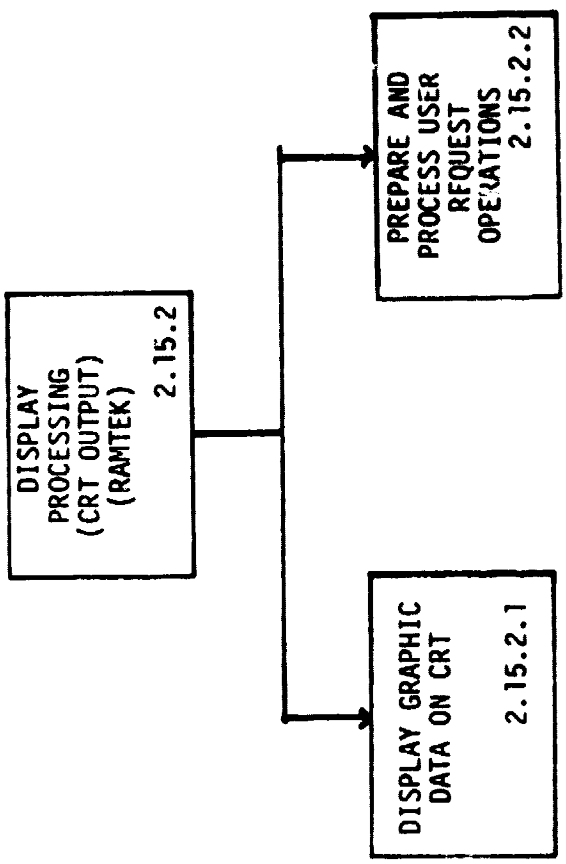

GENERATE COL-
OR REGISTERED
DISPLAY
2.13.3
CRTMAP
REGISTER
PIXELS AND
DISPLAY
2.13.3.1
CRTPRT

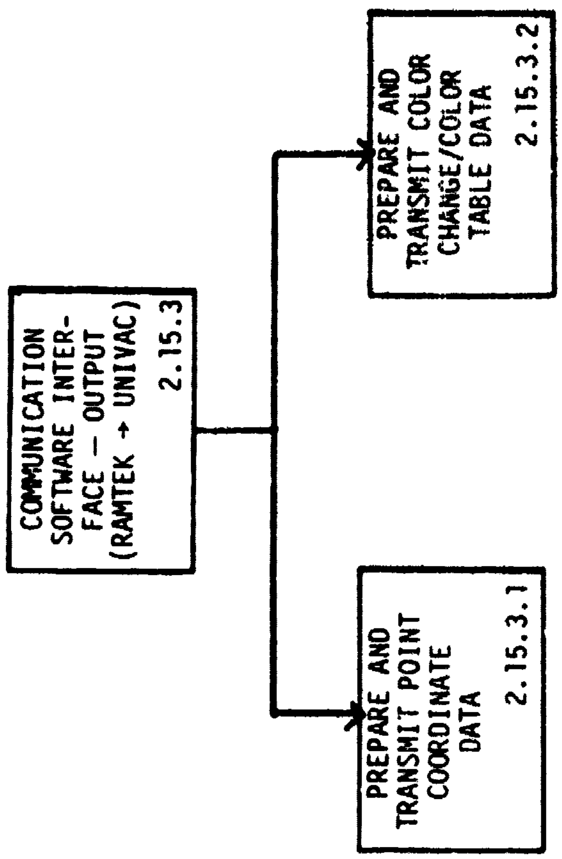

REGISTER
PIXELS AND
DISPLAY
2.13.3.1
CRTPRT
DISPLAY
TABULAR DATA
2.13.3.1.1
DISPLAY SCAN
LINE ON
CRT
2.13.3.1.2
CRTOUT


PROGRAM ANOMALIES 2.14
INITIALIZE FILES 2.14.1
BUILD SPECTRAL TABLE
COMMAND RADIUS 2.14.3
COMMAND ANOMLY 2.14.4


INTERACTIVE GRAPHICS SUBSYSTEM
2.15
COMMUNICATION SOFTWARE INTER-FACE — INPUT (UNIVAC → RAMTEK)
2.15.1
DISPLAY PROCESSING (RAMTEK)
2.15.2
COMMUNICATION SOFTWARE INTER-FACE — OUTPUT (RAMTEK → UNIVAC)
2.15.3


SOFTWARE COMMUNI-
CATION INTERFACE
— INPUT
(UNIVAC → RAMTEK)
2.15.1
DECODE UNIVAC
IMAGE DATA — PRE-
PARE AND TRANSMIT
RAMTEK DATA
2.15.1.1
DECODE UNIVAC
X-Y PLOT DATA
— PREPARE & TRANS-
MIT RAMTEK DATA
2.15.1.2
DECODE UNIVAC
TEXT (TABULATION)
DATA — PREPARE &
TRANSMIT RAMTEK
DATA
2.15.1.3


DISPLAY PROCESSING (CRT OUTPUT) (RAMTEK)
2.15.2
DISPLAY GRAPHIC DATA ON CRT
2.15.2.1
PREPARE AND PROCESS USER REQUEST OPERATIONS
2.15.2.2


COMMUNICATION SOFTWARE INTERFACE — OUTPUT (RAMTEK → UNIVAC)
2.15.3
PREPARE AND TRANSMIT POINT COORDINATE DATA
2.15.3.1
PREPARE AND TRANSMIT COLOR CHANGE/COLOR TABLE DATA
2.15.3.2

Author: ______ Date: 02/07/79

Diagra ID: 0.0 Name: RSIS Description: REMOTE SENSING INFORMATION SUBSYSTEM (production and demand modes)

Input	Process	Output
Source data tapes: - Landsat - Aircraft - Ground measurements Card images: - Ground measurements - Potential control points - User directives Source data disk files: - Detection data - Control points Validation File System files: - Program File - Run Information File - Default File	1. Initialization of a. User directives b. System files c. Program requests (1.0). 2. Processing of standard data inputs (2.0). 3. Output standard and special purpose data products (3.0).	Tape (classified or unclassified): - Landsat (copy or reformatted) - Aircraft (copy) - Plot CCT's - Special format Scaled Maps: - Pen plots - Line printer plots Tabulations: - Data Availability Reports - Data Update Reports - Special Computations - Control Network Reports - Detection File Reports - Scene ID Disk files: - Detection Files - Adjusted Control Point File Film output Printer Displays

Author: ______ Date: 02/07/79

Diagram ID: 1.0 Name: ______ Description: INITIALIZATION

Input	Process	Output
Validation File Card images: - Ground measurements - Potential control points - User directives System Files: - Program File - Run Information File - Default File Demand Switch*	1. Pick up and edit initialization directives (1.1). 2. If in production mode and any error occurred in pick up and edit, error exit. 3. Set up system files (1.2). 4. Set up controls to initialize requested programs (1.4).	Source Data Availability Report Card Image Printout Validation messages Ground Measurement File Potential Control Point File Validated user directives: - Input sources - Output products - Program choices - Output directives Run Stream File

*This switch is turned on when the user signs on the terminal.

Author: ______________________ Date: 02/07/79

Diagram ID: 1.1 Name: ______________________ Description: PICK UP AND EDIT

Input

Validation File:

- Valid input information
- Valid output information
- Valid program choice information

Card images:

- Ground measurements
- Potential control points
- User directives:

  Special calculation parameters

  Run option requests

  Output directives

  Input sources

  Program choices

  Output titles

. demand switch

Process

1. Pick up validation file information for validation purposes.
2. Get valid input source(s) (1.1.1).
3. Get valid output product(s) (1.1.3).
4. Get valid program choice(s) (1.1.2).
5. Get valid output directives (1.1.4).
6. Produce interpreted card listing.
7. If termination flag is set, exit.
8. If error occurred, repeat 2 through 7 as needed.

Output

Validation File:

. valid tape densities

. valid tape numbers

. valid tape sources

. valid input file names

. valid program names

. valid scene ID

Input sources:

. validated tape numbers

. validated file names

. validated data source/type

. validated number of tapes

Output products

Program choices

Output directives:

. validated output (LP, plots, tape, disk, terminal, film)

. validated run number

. validated output titles

. interactive switch (to override default)

Interpreted Card Listing

. termination flag

. error flag

Author:

Date: 02/07/79

Diagram ID: 1.1.1 **Name:** **Description:** GET VALID INPUT SOURCES

Input

. demand switch

Card images

Validation File:

. valid tape densitites

. valid tape numbers

. valid input types

. valid file names

. valid file types (control point network, radiance, class value, density, ground measurements, potential control points)

Process

1. If demand mode, output prompt.
2. Read card image.
3. If production mode, list card image.
4. Check validation file information to see if the type of input is valid.
5. Validate any tape numbers (1.1.1.1).
6. Validate any quantity of tapes and density (1.1.1.2).
7. Validate any file name (1.1.1.3).
8. Validate any ground measurements (1.1.1.4).
9. Validate any potential control points (1.1.1.5).
10. If termination is desired, set termination flag.
11. If batch and error, set error flag.
12. If demand mode, repeat steps 1 through 9 until correct information is stored, or termination.

Output

Card Image Printout

. types of input

. tape numbers

. quantities of tapes

. file names

. validated types of input

. validated tape numbers

. validated quantities of tapes

. validated file names

. validated file types

Ground Measurement Files

Potential Control Point Files

. density

. validated density

. termination flag

. error flag

Author: ______________ Date: 11/01/78

Diagram ID: 1.1.1.1 Name: ______________ Description: VALIDATE TAPE NUMBERS

Input	Process	Output
. tape number . validated type of input Validation file: . valid tape numbers . valid types of input	1. Find the tape number in the validation file information. 2. Verify that the type of input associated with that number in the validation file information is the same as the type of input requested. 3. If errors, print error message and set error flag.	. error flag Error messages

Author: ____________ Date: 02/07/79

Diagram ID: 1.1.1.2 Name: ____________ Description: VALIDATE QUANTITY OF TAPES

Input	Process	Output
. valid density . quantity of tapes . type of input . density	1. Compare quantity of tapes requested to maximum number permitted for that input type. a. If density is 800 bpi and type is Landsat, the maximum number of tapes is 4. b. If density is 1600 bpi and type is Landsat, the maximum is 2. c. For all other types of tapes, the maximum is 1. 2. If the quantity is larger than the above determined number, print error message and set error flag.	. error flag Error messages

Author: ______ Date: 11/01/78

Diagram ID: 1.1.1.3 Name: ______ Description: VALIDATE FILE NAME

Input	Process	Output
Validation File: . valid file names . valid file types . file name . file type: Control Point Network, Detection File (radiance, class value, density, Ground Measurements, Potential Control Points)	1. Find file name in validation file information. 2. Check if type of input associated with that file name in the validation file information is the same as the type of input specified. 3. If errors, set error flag and print error messages.	. error flag Error messages

Author: ______ Date: 11/03/78

Diagram ID: 1.1.1.4 Name: ______ Description: VALIDATE GROUND MEASUREMENTS

Input

Card Images:
- Ground measurements
- User directives

Validation File:
. valid file names
. valid file types

Process

1. If demand mode, enter file name or card image.
2. If file,
   a. Find file name in validation file information.
   b. Check to see that the type of file associated with that file name is a ground measurements file.
3. If card image:
   a. Create file
   b. Read and store ground measurements in new file until user indicates done.
   c. Make an entry concerning new file in validation file.
   d. Print each ground measurement.
4. If errors, print error messages and set error flag.

Output

Validated Ground Measurements File
. validated file name
. validated file type

Ground Measurement Printout

Error messages
. error flag

Author: ______

Date: 02/07/79

Diagram ID: 1.1.1.5 Name: ______ Description: VALIDATE POTENTIAL CONTROL POINTS

Input

. demand switch

Card images:

- Potential control points
- User directives

Validation File:

. valid file names

. valid file types

. valid tape numbers

Process

1. If demand mode, output prompt.
2. Read input card image.
3. If production mode, list card image.
4. If tape or file is input:

   a) Find tape number or file name in validation file information.

   b) Check to see that type of file (tape) associated with that file (tape) is a potential control point file (tape).

5. If cards are input:

   a) If demand mode, output prompt.

   b) Read input card image and store in new file until user indicates done.

   c) Print each potential control point.

6. If error, print error messages and set error flag.

Output

. validated file name

. validated file number

. validated file type

Validated Potential Control Points File

Potential Control Point Printout

Error message

. error flag

Author: ____________ Date: 11/03/78

Diagram ID: 1.1.2 Name: ____________ Description: GET VALID OUTPUT PRODUCTS

Input

. demand switch

Card images

Validation File:

. valid output products

. valid file names

. valid file types

. valid scene ID's

Process

1. If demand mode, output prompt.
2. Read card image.
3. If production mode, list card images.
4. Check validation file information to see if the type of output product is valid.
5. Validate any file name, if that output file is to be an update (1.1.1.3)
6. Pick up scene ID for this file from validation file and save.
7. If termination is desired, set termination flag.
8. If batch and errors, set error flag and print error messages.
9. Repeat 1 through 7 until correct output products are stored or termination.

Output

Card Image Printout

Validated output products:

. validated output types

. validated file names

. current scene ID

. termination flag

Error messages

. error flag

Author: ____________ Date: 11/01/78

Diagram ID: 1.1.3 Name: ____________ Description: GET VALID PROGRAM CHOICES

Input	Process	Output
Validation File . validated output choices . necessary programs versus output choices table . validated input choices . necessary inputs versus programs table	1. Generate list of all programs to be run to obtain output choices. 2. Check to see that all the necessary validated input choices are available for the required program choices. 3. If error, write error message and set error flag.	. programs list . validated program choices Error messages . error flag

Author: 　　　　　　　　　　　　　　　　Date: 11/02/78

Diagram ID: 1.1.4　　Name: 　　　　　　Description: GET VALID OUTPUT DIRECTIVES

Input	Process	Output
. demand switch Card image: - User directives: run number output titles other run options	1. If demand mode, output prompt. 2. Read input card image. 3. If production mode, list card image. 4. Check if run number is an alphanumeric field of up to 3 characters long, only 2 of which can be numeric. 5. Validate output titles, when applicable. 6. Validate other run options, when applicable. 7. If any error occurs, write error message and set error flag.	Card Image Printout . valid run number . valid output titles . other valid run options such as interactive switch Error message . error flag

Author: ______ Date: 10/02/78

Diagram ID: 1.2 Name: ______ Description: SET UP SYSTEM FILES

Input	Process	Output
. validated file names	1. Assign program file.	Error messages
System Files	2. Assign run information file.	Executive messages
- Program File	3. Assign default file.	
- Run Information File	4. Assign run stream file.	
- Default File		

Author:

Diagram ID: 1.4   Name: Date: 10/02/78

Description: SET UP CONTROLS TO INITIALIZE PROGRAM REQUESTS (building a run stream file)

Input

System Files

- Program File
- Run Information File
- Default File

. interactive mode switch

. valid file names

. valid number of tapes

. valid tape numbers

. valid program names

. valid output directives (line printer, plots, tape, disk, terminal)

Process

1. Add elements to Run Stream File from Run Information File.
2. If production mode, rename Program File.
3. If production mode, issue ADD commands for absolutes.
4. Merge input sources and Default File with Run Information File.
5. Validate run stream syntax.
6. If syntax error, correct
7. If fatal error, EXIT.
8. Repeat items ‘ through 7 until all correct or EXIT.

Output

Run Stream File (production)

TRP$ Run Stream (demand)

Error messages

. error flag

Author: ____________ Date: 03/09/79

Diagram ID: 2.0 Name: ____________ Description: PROCESS PROGRAMS

Input	Process	Output
Card images - User commands - Ground measurements - Potential control points Source data tapes - Landsat - Aircraft Source data files - Landsat - Aircraft Detection files - Class - Radiance - Density RSIS files - Validation file - Default files - Run information files	Process one of the following programs:	
	1. Edit RSIS files (2.1).	RSIS Files: - Validation Files - Default Files - Run Information Files
	2. Identify Landsat scene (2.2).	Scene ID
	3. Duplicate Landsat format tape, on tape or disk (2.3).	Source Data Tape - Landsat - Aircraft in Landsat format
	4. Reformat aircraft tapes to Landsat format (2.4).	Source Data Tape - Aircraft in Landsat format
	5. Adjust Control Point Network (2.5).	Adjustment Summary Residual Errors Report Adjusted Control Point Network File Control Network Diagrams

Author:

Diagram ID: 2.0 Name: Date: 03/09/79

Description: PROCESS PROGRAMS

Input	Process	Output

	6. Display Landsat tapes on line printer (2.6).	Displays
		- Radiance
		- Function of radiance
		  - Gradient
		  - Laplacian
		  - Class

	7. Display Landsat tapes on color CRT (2.7).	Tabulations Histograms Lengths Areas
	8. Classify data according to spectral limits (2.8).	Detection File - Density - Radiance - Class
	9. Print Detection File (2.9).	Display - Density - Radiance - Class List - Density - Radiance - Class

Author: ______ Date: ______

Diagram ID: 2.0 Name: ______ Description: PROCESS PROGRAMS

Input	Process	Output
	10. Produce registered line printer maps (2.10).	Maps Tabulations Histograms Lengths Areas
	11. Produce registered plot maps (2.11).	Plot maps Tabulations Histograms Lengths Areas
	12. Produce registered film displays (2.12).	JSC Universal Image Data Format tape to the Production Film Converter Tabulations Histograms Lengths Areas
	13. Produce registered color CRT displays (2.13).	Color Registered displays on CRT Tabulations Histograms Lengths Areas

Author: ______________

Date: ______

Diagram ID: 2.0

Name: ______________

Description: PROCESS PROGRAMS

Input	Process	Output
	14. Delete anomalies from Detection File (2.14).	Detection Files
	15. Interactive graphics subsystem (2.15).	Graphics records in RAMTEK format Image records in RAMTEK format CRT displays - Image - X-Y plot - Text Messages and data to UNIVAC

Author: ____________

Date: 02/08/79

Diagram ID: 2.1

Name: ____________

Description: EDIT RSIS FILES

Input

RSIS Files:
- Validation Files
- Default Files
- Run Information Files

→

Process

1. Use Univac text editor to edit contents of files:

   a. Add

   b. Update

   c. Delete

   d. Maintenance

→

Output

RSIS Files:
- Validation Files
- Default Files
- Run Information Files

Author:

Date: 01/19/79

Diagram ID: 2.2 **Name:** ERTSIDC **Description:** IDENTIFY LANDSAT TAPE

Input

Source data tape:

- Landsat

. data/checkout mode switch (MDATAC)

Process

1. Open tape file.
2. Do nothing further if in data/checkout mode.
3. Print ID information for Landsat scene on logical unit 6 (PRINT$file)
4. Put Landsat ID on tail sheet.
5. Terminate.

Output

Scene identification printouts

Author:

Date: 03/08/79

Diagram ID: 2.3 Name: ERTSDUP Description: DUPLICATE LANDSAT FORMAT SOURCE TAPE ON DISK OR TAPE

Input	Process	Output
Registration Parameter File	1. Initialize by loading system date and time, identifying program name, and setting mode switches (2.3.1).	. system date . system time . mode switches . registration parameters
Card image	2. Read commands and specifications, and save command name.	. command name
	3. If command defines or selects specifications, store them in applicable variables (2.3.2).	. specification variables
Source Data Tape: - Landsat - Aircraft in Landsat format	4. If command is DUPLICATE, copy Landsat tape to disk or tape (2.3.3).	Output (source) Data Tape: - Landsat - Aircraft in Landsat format Output (source) Data File: - Landsat - Aircraft in Landsat format
	5. If command is EXIT, exit from ERTSDUP (2.3.4).	
	6. If illegal command, print message.	Error messages
	7. Repeat from step 2.	

Author: ____

Date: 03/08/79

Diagram ID: 2.3.1 Name: ERTSXQT Description: INITIALIZE

Input	Process	Output
Source Data Tape: → - Landsat - Aircraft in Landsat format Registration Parameter File →	1. Get system date and time, set mode switches, print program start message and place message in Log File. → 2. Identify and print Landsat scene ID, build input window packets, and identify Landsat format used. → 3. Load registration parameters. If not available, load nominal registration parameters. → 4. Queue default commands.	Program start message . mode switches . system date . system time Log File Scene ID . scene ID . input window packet . type of Landsat format - old - new . registration parameters

Author: ______ Date: 03/08/79

Diagram ID: 2.3.2 Name: ______ Description: PROCESS COMMANDS THAT SPECIFY

Input

- command name

Process

1. If command is ORIGIN, store window origin (2.7.2.5).
2. If command is MERIDIAN, store non-standard UTM central meridian (2.10.2.3).
3. If command is WINDOW, store window vertices (2.7.2.9).
4. If command is ZONE, store UTM zone and standard UTM central meridian (2.7.2.10).

Output

- origin
- UTM central meridian
- window vertices
- UTM zone
- UTM central meridian

Author: ______ Date: 03/08/79

Diagram ID: 2.3.3.1 Name: ______ Description: COPY OLD FORMAT TAPE

Input

. output window

. registration parameters

. input window

Source Data Tape:

- Landsat
- Aircraft in Landsat format

Process

1. If registration parameters are not available and user has specified a window in GED or UTM coordinates, print message and return.
2. Read ID record.
3. Save strip number.
4. Copy ID record to output ID record buffer.
5. If output window is an extract of input window*:
   a. Calculate new data record length.
   b. Store new data record length on output ID record.
   c. Add flag at end of ID record indicating extract.
   d. Calculate retrieval envelopes in all coordinate systems and store in various window packets.
   e. Store window packets at end of output ID record buffer.

Output

Error message

. strip number

. output window packets

*The input window defines the area covered by the scene or strip on the input tape.

Author: ____

Diagram ID: 2.3.3 Name: ____

Date: 03/08/79

Description: COPY LANDSAT TAPE TO DISK OR TAPE

Input

Card image

. type of Landsat format

Process

1. Get name of user-specified output device. If not TAPE or DISK, print message and set error flag.
2. If bad scan line removal was requested, set bad scan line removal flag.
3. If threshold was specified, save. If not specified, save default threshold.
4. If error flag, return.
5. If old format tape, copy to tape or disk (2.3.3.1).
6. If new format tape, copy to tape or disk (2.3.3.2).

Output

Error messages

. error flag

. output device

- Tape
- Disk

. bad scan line removal flag

. threshold

Source Data Tape:

- Landsat or
- Aircraft in Landsat format

Author: ______ Date: 03/08/79

Diagram ID: 2.3.3.1 Name: ______ Description: COPY OLD FORMAT TAPE

Input	Process	Output
Source Data Tape - Landsat - Aircraft in Landsat format . input window . output window . strip number . bad scan line removal flag	6. Write output ID record buffer to output file. 7. Read annotation record. 8. Copy annotation record to output file. 9. If output window is an extract of input window: a. Determine first and last scan line to read that is inside window. b. Position tape at that scan line. 10. Read scan line. 11. Copy portions of scan line that fall within the output window to output scan line buffer. 12. If the bad scan line removal flag is on, if the strip number is 1 or 4, and if the first 24 bits of input scan line indicate a bad scan line, replace output scan line with previous output scan line, if available.	Output Data File: - ID record - Annotation record

Author: ______ Date: 03/08/79

Diagram ID: 2.3.3.1 Name: ______ Description: COPY OLD FORMAT TAPE

Input	Process	Output
. threshold . bad scan line removal flag . strip number Source Data Tape: - Landsat - Aircraft in Landsat format	13. If bad scan line removal flag is on, if the strip number is 2 or 3, and if the previous output scan line is available: a. Perform auto-correlation between the current and previous output scan lines (2.3.3.1.1). b. Apply threshold. c. If resulting value indicates bad output scan line, replace with previous output scan line. 14. If current output scan line is good, save it as previous output scan line. 15. Write output scan line to output file. 16. If this is last scan line to read, position tape at end of image data section if not already there, and skip to step 18. 17. Repeat from step 10. 18. If strip number is 4, read and copy all SIAT data to output file.	Output Data File: - Scan lines - SIAT data

Author: ______ Date: 03/08/79

Diagram ID: 2.3.3.1 Name: ______ Description: COPY OLD FORMAT TAPE

Input	Process	Output
	19. Write end-of-tape on output tape.	

Author: ______ Date: ______

Diagram ID: 2.3.3.1.1 Name: ______ Description: AUTO-CORRELATION OF TWO SCAN LINES

Input	Process	Output
	1. Not yet implemented.	

Author: ____________ Date: 03/08/79

Diagram ID: 2.3.3.2 Name: ____________ Description: COPY NEW FORMAT TAPE

Input	Process	Output
. output window . input window . registration parameters Source Data Tape: - Landsat - Aircraft in Landsat format	1. If registration parameters are not available and user has specified a window in UTM or GED coordinates, print message and return. 2. Read tape directory and write to output file. 3. Read header record and copy to output header buffer. 4. If output window is an extract of input window, calculate and store the image data record length on output header buffer. 5. Write output header buffer to output file. 6. Read first ancillary record and store in ancillary buffer. 7. If output window is an extract of input window, store number of pixels per line and number of lines in extract in ancillary buffer. 8. Write ancillary buffer to output file. 9. Read second ancillary record and write to output file.	Error messages Output Data File: - Tape directory - Header record - First ancillary record - Second ancillary record

Author: ____________

Date: 03/08/79

Diagram ID: 2.3.3.2

Name: ____________

Description: COPY NEW FORMAT TAPE

Input	Process	Output
. output window put window Source Data Tape: - Landsat - Aircraft in Landsat format . bad scan line removal flag	10. Read annotation record and write to output file. 11. In output window is extract of input window: a. Determine first and last scan line to read that is inside output window. b. Position tape at that scan line. 12. Read scan line. 13. Copy portions of scan line that fall in the output window to output scan line buffer. 14. If output window is an extract of input window, store number of pixels per scan line in output scan line buffer. 15. If bad scan line removal flag is on and the previous output scan line is available: a. Perform auto-correlation between the current and previous output scan lines (2.3.3.1.1).	Output Data File: - Annotation record

Author: ______

Diagram ID: 2.3.3.2 Name: ______

Date: 03/08/79

Description: COPY NEW FORMAT TAPE

Input	Process	Output
. threshold Source Data Tape: - Landsat - Aircraft in Landsat format	b. Apply threshold. c. If resulting value indicates bad output scan line, replace with previous output scan line. 16. If current output scan line is good, save it as previous output scan line. 17. Write output scan line to output file. 18. If this is last scan line to read, position tape at end of image data section, if not already there, and skip to step 20. 19. Repeat from step 12. 20. Read and copy trailer file to output file. 21. Write end-of-file on output tape.	Output Data File: - Scan lines - Trailer file

Author:

Date: 02/26/79

Diagram ID: 2.3.4 Name: ERTSEXI Description: EXIT FROM ERTSDUP

Input	Process	Output
. fatal error flag →	1. If fatal error occurred, exit in error. 2. Exit.	

Author:

Date: 03/08/79

Diagram ID: 2.4   Name: REFMT   Description: REFORMAT TAPES

Input	Process	Output
Source Data Tape: - Aircraft in aircraft format	1. Initialize by loading system date and time, printing program start message, and setting mode switches (2.4.1).	
	2. If command is CHANNEL, store user-specified channel numbers (2.7.2.2).	. channels selected
	3. If command is SCENE, store Landsat scene ID (2.4.3).	. scene ID
	4. If command is REFORMAT, reformat aircraft tape into Landsat format (new) (2.4.2).	Source Data Tape - Aircraft in Landsat format
	5. Exit from REFMT (2.4.4).	

Author:

Date: 03/08/79

Diagram ID: 2.4.1

Name: REFXQT

Description: INITIALIZE

Input	Process	Output
Aircraft tape	1. Get system date, time, and mode switches.	. system date . system time . mode switches
	2. Print program start message and place message in Log File.	Program start message
	3. Identify type of tape format.	Log File . format type
	4. If invalid, print message, set error flag, and exit in error (2.4.4).	Error message

Author: ______

Diagram ID: 2.4.2 Name: ______ Date: 03/08/79

Description: REFORMAT AIRCRAFT TAPE

Input

. channels specified

. type of format

. scene ID

Process

1. If number of channels specified is not four, print message and set error flag.
2. Validate channel numbers:
   a. If type of aircraft format is high altitude MS, check that each channel number specified is not larger than five. If invalid, print message and set error flag.
   b. If type of aircraft format is MS, check that each channel number specified is not larger than eight. If invalid, print message and set error flag.
   c. If type of aircraft format is MSS, check that each channel number specified is not larger than 11. If invalid, print message and set error flag.
3. If Landsat scene ID is not specified, print message and set error flag.
4. If error flag is on, return.

Output

Error messages

. error flag

Author: ____________ Date: 03/08/79

Diagram ID: 2.4.2 Name: ____________ Description: REFORMAT AIRCRAFT TAPE

Input	Process	Output
. format type - RM-18MS (High Altitude Imagery Universal Format Raw Data) - N5001MS (Imagery Universal Format Raw Data) - MMS (MMS Imagery Universal Format Raw Data)	5. Define byte assignment matrix depending on type of format*. 6. Open image data file. 7. Reformat aircraft tape to Landsat format (2.4.2.1).	. byte assignment matrix Image Data File Source data tape - Aircraft data in Landsat format

*The three aircraft formats contain the same information in nearly the same locations. Most of the differences among the byte assignments for the various formats are caused by the different numbers of channels each supports.

Author: ______

Date: 03/08/79

Diagram ID: 2.4.2.1

Name: ______

Description: REFORMAT AIRCRAFT TAPE

Input

Source data tape

- Aircraft (MS)

. byte assignment matrix[2]

Process

1. Read aircraft tape header record.
2. Use information in aircraft header record to build Landsat format[1] tape directory and write to output tape (2.4.2.1.1).
3. Use information in aircraft header record to build Landsat header record and write to output tape (2.4.2.1.2).
4. If current channel is not one of the user-specified channels, position tape at end of current channel and skip to step 7.
5. Read image data for all scan lines for the current channel[3], and store in Image Data File.
6. If this is first channel stored, calculate and save the number of scan lines.
7. Repeat from step 4 for each channel on aircraft tape.
8. Rewind and release aircraft tape.

Output

. aircraft tape header record

Aircraft tape in Landsat format:

- Tape directory

- Header record

Image Data File

. number of scan lines

[1] Landsat III version 0.0 format.

[2] The byte assignment matrix contains the information used by this module and the ones below it to know the exact

Author:

Date: 03/08/79

Diagram ID: 2.4.2.1

Name:

Description: REFORMAT AIRCRAFT TAPE

Input	Process	Output
. byte assignment matrix	9. Use information in aircraft header record to build Landsat format ancillary records, and write to output tape (2.4.2.1.3). 10. Use information in aircraft header record to build Landsat format annotation record and write to output tape (2.4.2.1.4). 11. Use information in Image Data File to build image data records in Landsat format interleaved by line and write to output tape (2.4.2.1.5). 12. Build Landsat format trailer file and write to output tape (2.4.2.1.6). 13. Delete image data file.	Aircraft tape in Landsat format: - Ancillary records - Annotation record - Image data - Trailer file

[3] Image data in this format is interleaved by channel: All scan lines ch. 1; all scan lines ch. 2 ... all scan lines ch. n.

Author: ____________ Date: 03/08/79

Diagram ID: 2.4.2.1.1 Name: ____________ Description: BUILD LANDSAT TAPE DIRECTORY

Input

. aircraft tape header

. byte assignment matrix

. scene ID

Process

1. Using the byte assignment matrix get the following from the header record and store as tape ID:

   a. Store aircraft mission number as mission number.

   b. Store aircraft sensor ID as sensor type.

   c. Store constant information.

2. Store tape ID in tape directory buffer.

3. Get and store date of tape creation in tape directory buffer.

4. Get and store aircraft tape sequence number as CCT volume number in the tape directory buffer.

5. Get and store computing system ID as site of CCT production in tape directory buffer.

6. Store Landsat scene ID in tape directory buffer.

7. Store constant information in tape directory buffer.

Output

Author:

Date: 03/08/79

Diagram ID: 2.4.2.1.1 Name:

Description: BUILD LANDSAT TAPE DIRECTORY

Input	Process	Output
	8. Write tape directory buffer to output tape.	Tape directory: - Tape ID - Date of tape creation - CCT volume number - Site of CCT production - Scene ID - Other tape directory fields

Author:

Date: 03/09/79

Diagram ID: 2.4.2.1.2

Name:

Description: BUILD LANDSAT HEADER RECORD

Input

. aircraft tape header record

. byte assignment matrix

Process

1. Using the byte assignment matrix, get the following from the header record and store as image ID:

   a. Store aircraft date and time information as spacecraft date and time.

   b. Store constant information.

2. Store image ID in header record buffer.

3. Get and store the following as spacecraft description:

   a. Store aircraft sensor ID as sensor ID.

   b. Store aircraft mission number as mission number.

   c. Store active channel flags as active detector status word.

   d. Store number of aircraft channels as number of active detectors.

   e. Store number of video elements per scan as number of pixels per scan line.

Output

Author:

Date: 03/08/79

Diagram ID: 2.4.2.1.2

Name:

Description: BUILD LANDSAT HEADER RECORD

Input

. aircraft tape header record

. byte assignment matrix

Process

4. Store spacecraft description in header record buffer.

5. Get and store aircraft date and time information as time scene was exposed, in header record buffer.

6. Get and store in header record buffer the sizes (in bytes) of various record types; the number of ancillary, trailer, and annotation records; various physical record sizes; and the number of lines and channels per physical record.

7. Get and store number of bits per pixel in header record buffer.

8. Store constant information in header record buffer.

9. Store flag indicating extract at end of header record buffer.

10. Write header record buffer to output tape.

Output

Header record:

- Image ID
- Spacecraft description
- Time of exposure
- Header, annotation, ancillary, image data, and trailer record lengths
- Number of ancillary annotation and trailer records
- Physical record sizes
- Number of lines per physical record
- Number of channels per physical record
- Number of bits per pixel
- Other header record fields
- Extract flag

C-2

Author: ______ Date: 03/08/79

Diagram ID: 2.4.2.1.3 Name: ______ Description: BUILD LANDSAT ANCILLARY RECORDS

Input	Process	Output
. aircraft tape header record . number of scan lines . byte assignment matrix	1. Using the byte assignment matrix, get the following and store in first ancillary record buffer: a. Store number of video elements per scan as number of pixels per scan line. b. Store number of scan lines as number of lines in extract. c. Calculate number of horizontal and vertical meters per pixel from aircraft velocity and altitude. d. Store aircraft altitude as spacecraft altitude. e. Calculate image width in meters from meters per pixel between pixels and number of pixels per scan. f. Store angle of areas mirror angle. g. Store sensor scan rate as time of active sweeps. h. Store constant information.	

Author:

Date: 03/08/79

Diagram ID: 2.4.2.1.3 Name:

Description: BUILD LANDSAT ANCILLARY RECORDS

Input

. aircraft tape header record

. byte assignment matrix

Process

2. Write first ancillary record buffer to output tape.

3. Get and store the following in second ancillary record buffer:

   a. Store aircraft angle of drift as spacecraft heading angle.

   b. Store ground speed in meters per second as spacecraft velocity.

   c. Store altitude in meters as spacecraft altitude.

   d. Store constant information.

4. Write second ancillary record buffer to output tape.

Output

First ancillary record:

- Number of pixels per scan line
- Number of scan lines
- Meters per pixel between pixels
- Meters per pixel between lines
- Spacecraft altitude
- Image width in meters
- Mirror angle
- Time of active sweeps
- Other ancillary fields

Second ancillary record:

- Spacecraft heading angle
- Spacecraft velocity
- Spacecraft altitude
- Other ancillary fields

Author:

Diagram ID: 2.4.2.1.4

Name:

Date: 03/07/79

Description: BUILD LANDSAT ANNOTATION RECORD

Input	Process	Output
. scene ID	1. Store Landsat scene ID in annotation record buffer. 2. Store constant information in annotation record buffer. 3. Write annotation record buffer to output tape.	Annotation record: - Scene ID - Other annotation fields

Author: ______ Date: 03/08/79

Diagram ID: 2.4.2.1.5 Name: ______ Description: BUILD IMAGE DATA RECORDS

Input

. byte assignment matrix

. aircraft tape header record

Image Data File

Process

1. Set scan line number to zero.
2. Increment scan line number and store in scan line buffer.
3. Using byte assignment matrix, get and store number of aircraft pixels per scan as number of pixels in this scan line, in scan line buffer.
4. Get and store gains and biases applied to each channel in scan line buffer.
5. Store constant information in scan line buffer.
6. Read pixels for one scan line for one channel from Image Data File and store in scan line buffer.
7. Repeat from step 6 until all four channels for this scan line in Image Data File have been read and copied.

Output

Author: ______

Date: 03/08/79

Diagram ID: 2.4.2.1.5

Name: ______

Description: BUILD IMAGE DATA RECORDS

Input	Process	Output
. number of scan lines →	8. Write scan line buffer to output tape. →	Image data: - Scan line number - Number of pixels in this scan - Gains and biases applied - Other image data fields excluding the image data itself - Image data interleaved by line
	9. Repeat from step 2 for every scan line in image.	

Author: ______ Date: 03/07/79

Diagram ID: 2.4.2.1.6 Name: ______ Description: BUILD TRAILER FILE

Input	Process	Output
	1. Store constant information on trailer buffer. 2. Write trailer buffer to output tape. →	Trailer file

Author: ____

Date: 03/06/79

Diagram ID: 2.4.3

Name: KMDSCE

Description: COMMAND SCENE

Input	Process	Output
Card image	1. Get and check scene number. If invalid, print message and set error flag.	. error flag Error messages . scene number
	2. Get number of samples per scene.	. number of samples per scene
	3. If number of samples per scene is less than 900, it indicates the number of samples per strip, so multiply it by four. If invalid, print message and set error flag.	
	4. Check test of the card image. If not blank, print message.	
	5. If error flag, restore scene number and number of samples per scene.	
. confirm switch	6. If confirm switch is on, print scene number and number of samples per scene.	Scene number Number of samples per scene

Author:

Diagram ID: 2.4.4 Name: REFEXI Date: 03/07/79 Description: EXIT FROM REFMT

Input	Process	Output
• number of fatal errors (NDFATL) →	1. If a fatal error has occurred, exit in error. 2. Exit.	

Author:

Date: 01/17/79

Diagram ID: 2.5 Name: CONTROL Description: PROGRAM CONTROL

Input	Process	Output
Card image	1. Initialize control by calling CONXQT (2.5.1) 2. Get valid command from user. 3. If 'ADJ' call CONADJ (2.5.2) If 'DIA' call CONDIA (2.5.3) If 'EXI' call CONEXI (2.5.4) If 'ATT', 'CLE', 'EXP', 'NEW', 'NEX', 'OFF', 'ONØ', 'POI', 'POØ', 'PØØ', 'SCE', 'ZON', or '$AD', execute appropriate command.	

Author:

Date: 11/28/78

Diagram ID: 2.5.1   Name: CONXQT   Description: INITIALIZE CONTROL

Input	Process	Output
	1. Identify the program.	Printout of program name
	2. Flag pitch and roll as unknown.	. pitch in degrees (PITDEG) . roll in degrees (ROLDEG)
	3. Queue default commands.	

Author:

Date: 10/31/78

Diagram ID: 2.5.2

Name: CONADJ

Description: ADJUST CONTROL NETWORK

Input

Potential control points
- scene number
- number of samples per scene
- satellite pitch
- satellite roll

Process

1. Check for valid scene and attitude, and enough control points (2.5.2.1)
2. Determine which points to print residual errors for (2.5.2.2)
3. Initialize (2.5.2.3)
4. Estimate control network centroid, coverage (%), and STM central meridian (2.5.2.4)
5. Save nominal altitude (2.5.2.5)
6. Estimate, by use of a model, corrected line and sample numbers and STM coordinates (2.5.2.6)
7. Estimate forward STM coefficients (2.5.2.7)
8. Compute final altitude and projection central meridian (2.5.2.8)
9. Compute final corrected line and sample numbers, and STM coordinates (2.5.2.6)
10. Compute final forward and inverse STM coefficients (2.5.2.9)

Output

Author:

Diagram ID: 2.5.2 | Name: CONADJ | Date: 10/31/78 | Description: ADJUST CONTROL NETWORK (continued)

Input	Process	Output
	11. Compute STM/geographic coordinates of scene center and nadir (2.5.2.10)	
	12. Compute STM/geographic coordinates of control network centroid (2.5.2.11)	
	13. Compute difference between scene center and control network centroid (2.5.2.12)	
	14. Compute angle of rotation from STM to SRM (2.5.2.13)	
	15. Compute mapping coefficients in meters (2.5.2.14)	
	16. Print adjustment summary (2.5.2.15)	Adjustment Summary
	17. Compute and print residual errors (2.5.2.16)	Residual Errors Report
	18. Store registration parameters (2.5.2.17)	Registration Parameter File (coefficients of transformation from scanner-oriented to earth-oriented coordinate systems which gives the best fit.)
	19. Restore nominal altitude (2.5.2.18)	

Author: ______

Diagram ID: 2.5.2.1 Name: ______

Date: 11/17/78

Description: CHECK FOR VALID SCENE AND ATTITUDE

Input

Satellite information (NERTS):

- ERTS number
- days since launch
- GMT

- Number of samples/scene (NERSAM)
- pitch in degrees (PITDEG)
- roll in degrees (ROLDEG)
- number of control points (NCTLPT)
- batch flag (MBATCH)

Process

1. If scene not specified, print message.
2. If samples/scene not specified, print message.
3. If attitude (pitch and roll) not specified, print message.
4. If not enough control points, print message.
5. If not enough control points and batch, print fatal error message.

All but the last (#5) are non-fatal, as those values are initialized later.

Output

Warning messages

Error messages

Author:

Date: 01/17/79

Diagram ID: 2.5.2.2 Name:

Description: DETERMINE POINTS TO PRINT RESIDUAL ERRORS FOR

Input

User directives:

. diagram command (NODT)

Process

1. Initialize NODT to 'NON'.
2. Get NODT from storage.
3. If NODT = 'CON' set NPRPOS = 1 and NPRNEG = 0. Indicates diagram control points only.
4. If NODT = 'CHE', set NPRPOS = 0 and NPRNEG = 1. Indicates diagram check points only.
5. If NODT = 'ALL', set NPRPOS = 1 and NPRNEG = 1. Indicates diagram all points.
6. If NODT = 'NON', set NPRPOS = 0 and NPRNEG = 0. Indicates diagram no points.
7. If none of above, error.
8. If no control points and NODT = 'CON', error.
9. If no check points and NODT = 'CHE', error.

Output

Error messages

. NPRPOS

. NPRNEG

Author: ____________ Date: 01/17/79

Diagram ID: 2.5.2.3.1 Name: ____________ Description: ASSIGN SATELLITE SENSOR CONSTANTS

Input

Satellite information (NERTS):

. ERTS number

. days since launch

. GMT

→

Process

1. Assign arbitrary constants. All are numeric but NERSAT which = 'LANDSAT' and NERSEN = 'MSS'
2. If bad ERTS number, assign constants for ERTS-1 anyway.

→

Output

Satellite/sensor constants:

. nominal lines/scene (NERLIN)

. nominal samples/scene (NERSAM)

. satellite type (NERSAT)

. sensor type (NERSEN)

. nominal altitude in KM (ALTKM)

. nominal altitude in samples (ALTSAM)

Mirror scan values:

. NERCHA

. SCNA

. SCNB

. SCNC

. SCNW

. SCNTHS

. SCNITS

. ROLRAC

. scene center line (CTRLIN)

. scene center sample (CTRSAM)

. half of active scan angle (SCNTHA)

. radians per sample (SCNTIW)

Author: ______________________ Date: 11/17/78

Diagram ID: 2.5.2.3 Name: ______________________ Description: INITIALIZE

Input

. window number (NWNDOW)

. satellite roll in degrees (ROLDEG)

Satellite information (NERTS):

. ERTS number

. days since launch

. GMT

Process

1. NWNDOW = IABS (NWNDOW) CONTROL uses no alternate print files, so it is not concerned about opening them. When NWNDOW is positive, the file(s) have been opened.
2. Assign satellite/sensor constants (2.5.2.3.1)
3. Calculate the satellite roll in radians.
4. NWNDOW = IABS (NWNDOW)
5. Print headings (2.5.2.3.2)
6. Print (ERTS) scene number, days since launch, GMT.

Output

. satellite roll in radians (ROLRAD)

. window number (NWNDOW)

Satellite/sensor constants:

. nominal lines/scene (NERLIN)

. nominal samples/scene (NERSAM)

. nominal altitude in KM (ALTKM)

. nominal altitude in samples (ALTSAM)

Mirror scan values:

. NERCHA

. SCNA

. SCNB

. SCNC

. SCNW

. SCNTHS

. SCNITS

. ROLRAC

. scene center line (CIRLIN)

. scene center sample (CTRSAM)

. half of active scan angle (SCNTHA)

. radians per sample (SCNTIW)

Page heading output

Author: ______ Date: 11/17/78

Diagram ID: 2.5.2.3.2 Name: ______ Description: PRINT HEADINGS

Input

- output unit (NUNIT)
- last heading line to write (LASTHD)
- batch flag (MBATCH)
- top margin provided (MLTOP6) (1 or 0)
- run heading (JRPIDT)
- page counter (PAGE)
- window number (NWNDOW)
- alternate print file (NIT)
- heading lines (JHDG)

→

Process

1. Eject page and provide top margin.
2. Print single line heading or

   Print multi-line heading.
3. Skip lines after heading.

→

Output

Page heading output

Author: ______

Date: 11/17/78

Diagram ID: 2.5.2.4

Name: ______

Description: ESTIMATE CENTROID, COVERAGE, AND STM CENTRAL MERIDIAN

Input

- centroid line (CTDLIN)
- centroid sample (CTDSAM)
- centroid latitude (CTDLAT)
- centroid longitude (CTDLON)
- number of control points, check points, etc. (NETHI)

Parameter set to a value of either 1 or 2 at compile time. Constancs may not be changed:

- WLIN (1)
- WSAM (2)
- WLAT (1)
- WLON (2)

- number of control points (NCTLPT)
- percent of scene covered (PCTCTL)
- array containing line and sample coordinates for each point (ADJNET)
- latitude, longitude coordinates for each point (GEDNET)
- point numbers (NETPT)

Process

1. Set centroid line, sample, latitude, and longitude to zero.
2. Sum up the lines, samples, latitudes, and longitudes for each control point in the network. Skip check points which are negative.
3. Divide those sums by the number of control points and store as centroid line, sample, latitude, and longitude.
4. STMCMD = CTDLON
5. Calculate the percent of the scene covered by the network (2.5.2.4.1)

Output

- CTDLIN
- CTDSAM
- CTDLAT
- CTDLON

- STM central meridian (STMCMD)

- percent coverage of scene (PCTCTL)

Author: ____________

Date: 11/17/78

Diagram ID: 2.5.2.4.1

Name: PCTCOV

Description: PERCENT OF SCENE COVERED BY THE NETWORK

Input	Process	Output
. number of nodes in network (NETHI) . line and sample numbers for every control point . WLIN (1) . WSAM (2) . line at scene center (CTRLIN) . sample at scene center (CTRSAM)	1. Initialize the minimum and maximum values for the short and long axis for two envelopes. 2. Determine eight possible envelopes for the network. 3. Find the smallest long and short axis among the envelopes (not necessarily belonging to the same envelope). 4. Find the largest long and short axis. 5. Compare the ellipse that has the short and long axis equal to the mean of the maximum and minimum found above, to the whole scene. The percent of the scene that the ellipse covers is the percentage of the scene covered by the network.	. percent of the scene covered by the network (PCTCTL)

Author:

Diagram ID: 2.5.2.5 Name:

Date: 11/13/78

Description: SAVE NOMINAL ALTITUDE

Input

- number of samples/scene (NERSAM)
- calculated altitude in KM (ALTKM)
- roll at scene center in radians (ROLRAD)
- scan angle in radians (SCNTH2)

Process

1. Calculate number of samples/KM.
2. Calculate the nominal altitude.

Output

- samples per kilometer (SAM1KM)
- nominal altitude (ALTNOM)

Author:

Date: 11/17/78

Diagram ID: 2.5.2.6 Name: TAPCOR Description: ESTIMATE CORRECTED LINE AND SAMPLE NUMBERS, AND STM COORDINATES

Input

- altitude in KMs (ALTKM)
- samples/KM (SAM1KM)
- number of nodes (NETH1)
- WLIN (1)
- WSAM (2)
- line and sample numbers for each point (ADJNET)

Satellite information (NERTS):

- ERTS number
- days since launch
- GMT

- latitude and longitude for each point (GEDNET)
- easting and northing for each point (STMNET)
- WEA (1)
- WNO (2)
- STM central meridian (STMCMD)

Process

1. Calculate the altitude in samples.
2. For each point in the network, calculate the corrected line and sample from adjusted.
3. If GMT for this satellite is <0, then the corrected sample is the same as the adjusted sample.
4. Calculate the UTM coordinates for these points, from the geographic coordinates which you already have (2.5.2.6.1)

Output

- corrected line and sample numbers (CORNET)
- STM coordinates for each point (STMNET)
- altitude in samples (ALTSAM)

Author:

Date: 11/22/78

Diagram ID: 2.5.2.6.1

Name: G2U

Description: CALCULATE UTM FROM GEOGRAPHIC

Input	Process	Output
. degrees of latitude (DEGLAT) . degrees of longitude (DEGLON) . central meridian in degrees (CMDEG) . cosine of latitude in radians (COSO1) . cosine squared of latitude in radians (COSO2) . sine times cosine of latitude in radians (SINCOS) . eccentricity factor (ECCFAC)	1. Define constants. 2. If this routine has been called recently for the same latitude (and not been overlayed), then COSO1, COSO2, SINCOS, and ECCFAC are still available and do not need to be calculated again, even though they were local variables. 3. If not step 2, then calculate the COS, and COS^2 of the latitude in radians, as well as the sine times cosine of the latitude, and an eccentricity factor. 4. Compute easting. 5. Compute northing.	. cosine (COSO1) . $cosine^2$ (COSO2) . sine times cosine (SINCOS) . eccentricity factor (ECCFAC) . easting (UTME) . northing (UTMN)

Author:

Date: 11/22/78

Diagram ID: 2.5.2.7

Name: DLSTSQ

Description: ESTIMATE FORWARD STM COEFFICIENTS

Input

. number of control points (NODES)

. point numbers (IPT)

. corrected line for first node (X)

. corrected sample for first node (Y)

. easting for first node (P)

. northing for first node (Q)

Process

1. Calculate the coefficients of transformation for corrected to STM by setting up simultaneous equations in matrix H and calling Univac math pack routine DGJR that solves simultaneous equations.

Output

. array coefficients of transformation from corrected to STM (A)

Author: ______

Date: 11/28/78

Diagram ID: 2.5.2.8

Name: ______

Description: COMPUTE FINAL ALTITUDE AND PROJECTION CENTRAL MERIDIAN

Input

- altitude in KMs (ALTKM)
- samples/KM (SAM1KM)
- STM northing from 2 corrected coordinates (STMN4C)
- number of lines in full scene (NERLIN)
- roll at scene center in radians (ROLRAD)
- half of active scan angle in radians (SCNTHS)
- number of samples in full scene (NERSAM)
- line of scene center (CTRLIN)
- sample of scene center (CTRSAM)
- center latitude (CTRLAT)
- center longitude (CTRLON)

Process

1. Calculate the STM northing from the corrected center line, first sample.
2. Calculate the STM easting for same.
3. Calculate the STM northing from corrected center line, last sample.
4. Calculate the STM easting for same.
5. Calculate the base of triangle in KMs.
6. Calculate the altitude by knowing 2 angles and a side.
7. Calculate the samples/KM.
8. Calculate the geographic coordinates of the scene center from scanner coordinates.
9. Calculate the STM central meridian.

Output

- altitude in KM (ALTKM)
- samples/KM (SAM1KM)

Geographic coordinates of scene center:

- CTRLAT
- CTRLON

- STM central meridian (STMCMD)

Author: ______

Date: 11/28/78

Diagram ID: 2.5.2.9

Name: ______

Description: COMPUTE FINAL FORWARD AND INVERSE STM COEFFICIENTS

Input

- number of nodes (NETHI)
- point numbers (NETPT)
- corrected line and sample numbers for each node (CORNET)
- WLIN (1)
- WSAM (2)
- STM line and sample numbers for each node (STMNET)
- WNO (1)
- WEA (2)
- corrected to STM coefficients of transformation (CORSTM)
- STM to COR coefficients of transformation (STMCOR)

Process

1. Compute final forward corrected to STM coefficients of transformation (2.5.2.7)
2. Compute final STM to corrected coefficients of transformation (subroutine REVERT, external) (2.5.2.9.1)

Output

- STM to corrected coefficients of transformation (STMCOR)
- corrected to STM coefficients of transformation (CORSTM)

Author:

Date: 11/28/78

Diagram ID: 2.5.2.9.1

Name: REVERT

Description: CALCULATE REVERSE COEFFICIENTS OF TRANSFORMATION

Input

- an array dimensional to six places, containing one-way coefficients of transformation (A)

Process

Calculate equation 2 from equation 1

BX = A(1)*AX + A(2)*AY + A(3)

BY = A(4)*AX + A(5)*AY + A(6)

(equation 1)

AX = B(1)*BX + B(2)*BY + B(3)

AY = B(4)*BX + B(5)*BY + B(6)

(equation 2)

Output

- an array dimensional to six places containing reverse coefficients of transformation (B)

Author:

Diagram ID: 2.5.2.10 Name:

Date: 11/28/78

Description: COMPUTE STM/GEOGRAPHIC COORDINATES OF SCENE CENTER AND NADIR

Input	Process	Output
<u>Functions</u> STM northing from corrected (STMN4C) STM easting from corrected (STME4C) Corrected sample fror adjusted (CORS4A) <u>Variables</u> Scanner coordinates for scene center: . CTRLIN . CTRSAM Geographic coordinates for scene center: . CTRLAT . CTRLON . STM central meridian (STMCMD)	1. Calculate scene center easting and northing. 2. Calculate geographic coordinates of scene center from UTM (2.5.2.10.1)	. scene center easting (CTRE) . scene center northing (CTRN) Scene center geographic coordinates: . CTRLAT . CTRLON

Author: ______

Diagram ID: 2.5.2.10.1 Name: U2G

Date: 11/28/78

Description: CALCULATE GEOGRAPHIC COORDINATES FROM UTM

Input	Process	Output
. UTM easting (UTME) . UTM northing (UTMN) . central meridian (CMDEG) Values saved from the last time this routine was called. If not overlaid they are still available: . UTMEL . Q = (UTME - 500,000)/100,000 . $Q2 = Q^2$. $Q4 = Q^4$	1. If easting this time is same as last time this was called, do not calculate UTMEL, Q, Q2, or Q4. 2. Calculate UTMEL, Q, Q2, and Q4. 3. Calculate AT 4. Calculate ON	. UTMEL . Q . QZ . Q4 . latitude (AT) . longitude (ON)

Author: ______________________ **Date:** 11/28/78

Diagram ID: 2.5.2.11 **Name:** ______________________ **Description:** COMPUTE STM/GEOGRAPHIC COORDINATES OF CONTROL NETWORK CENTROID

Input	Process	Output
<u>Functions</u> STM northing from corrected (STMN4C) STM easting from corrected (STME4C) Corrected sample from adjusted (CORS4A) <u>Variables</u> . centroid line (CTDLIN) . centroid sample (STDSAM) . latitude for centroid (CTDLAT) . longitude for centroid (CTDLON) . STM central meridian (STMCMD)	1. Calculate centroid northing. 2. Calculate centroid easting. 3. Calculate geographic coordinates of centroid (2.5.2.10.1)	. easting centroid (CTDE) . northing centroid (CTDN) . centroid latitude (CTDLAT) . centroid longitude (CTDLON)

Author: ____________ Date: 11/28/78

Diagram ID: 2.5.2.12 Name: ____________ Description: COMPUTE DIFFERENCE BETWEEN SCENE CENTER AND CONTROL NETWORK CENTROID

Input	Process	Output
Center and centroid line: . CTRLIN . CTDLIN Center and centroid sample: . CTRSAM . CTDSAM Latitude: . CTRLAT . CTDLAT Longitude: . CTRLON . CTDLON Easting: . CTRE . CTDE Northing: . CTRN . CTDN	1. Calculate the line difference. 2. Calculate sample difference. 3. Calculate latitude difference. 4. Calculate longitude difference. 5. Calculate the difference in the centers in KMs.	. difference in line (DCLIN) . difference in sample (DCSAM) . difference in latitude (DCLAT) . difference in longitude (DCLON) . difference in center (DCKM)

Author: ____

Diagram ID: 2.5.2.13 Name: ____ Date: 11/28/78

Description: COMPUTE ANGLE OF ROTATION FROM STM TO SRM

Input	Process	Output
· coefficients of transformation for corrected to STM (CORSTM)	1. Calculate the angle of rotation using CORSTM coefficients.	· angle of rotation in radians (ROTRAD)

Author:

Diagram ID: 2.5.2.14

Name:

Date: 11/28/78

Description: COMPUTE MAPPING COEFFICIENTS IN METERS

Input	Process	Output
. coefficients of transformation for corrected to STM (CORSTM) . angle of rotation for STM to SRM (ROTRAD)	1. Calculate corrected to SRM coefficients of transformation.	. corrected to SRM coefficients of transformation (CORSRM)

Author: ____________ Date: 11/28/78

Diagram ID: 2.5.2.15 Name: ____________ Description: PRINT ADJUSTMENT SUMMARY

Input	Process	Output
Scene center coordinates: . CTRLIN . CTRLAT . CTRLON Centroid coordinates: . CTDLIN . CTDSAM . CTDLAT . CTDLON Difference between CTR and CTD: . DCLIN . DCSAM . DCLAT . DCLON . DCKM . STM central meridian (STMCMD) . percent of scene covered by nodes (PCTCTL) . nominal altitude (ALTNOM) . calculated altitude (ALTKM)	1. Print adjustment summary	Adjustment Summary

Author: ______________ **Date:** 11/28/78

Diagram ID: 2.5.2.16 **Name:** ERRORS **Description:** COMPUTE AND PRINT RESIDUAL ERRORS

Input	Process	Output
. root mean square error (meters) of adjustment (RMSMET) . number of nodes (NETHI) . report on control points (0 or 1) (NPRPOS) . point numbers (NETPT) . report on check points (0 or 1) (NPRNEG) . coordinates transformation (CORNET) . WLIN (1) . WSAM (2) . function (CORL4S) . STM coordinates transformation (STMNET) . WEA (1) . WNO (2) . functions (STME4C, STMN4C) . STM control meridian (STMCMD) . number of control points (NCTLPT) . geographic coordinates of nodes (GEDNET)	1. Initialize 2. Compute root mean square error	. root mean square error (RMSMET)

Author:

Date: 01/17/79

Diagram ID: 2.5.2.17

Name: STREG8

Description: STORE REGISTRATION PARAMETERS

Input	Process	Output
. window number (NWNDOW) . root mean square error (RMSMET) . percent of scene covered by control network (PCTCTL) Name list statements that include all data to be saved: . (LSTNER) . (LSTORB) . (LSTSCN) . (LSTFIT) Data included: NERTS, NERLIN, NERSAM, ALTKM, ALTSAM, CTRLIN, CTRSAM, CTRLAT, CTRLON, DIRLAT, DIRLON, PITDEG, ROLDEG, YAWDEG, SCNA, SCNB, SCNC, ROLRAC, SCNTHS, SCNTIW, NCTLPT, PCTCTL, RMSMET, UTMCMD, STMCMD, CORSTM, STMCOR, CORSRM	1. Check validity of control network adjustment. If inacceptable, print message and flag error on Registration Parameter File. 2. Write registration parameters on Unit 8 in name list form.	Error messages Registration Parameter File

Author: ______ Date: 11/13/78

Diagram ID: 2.5.2.18 Name: ______ Description: RESTORE NOMINAL ALTITUDE

Input	Process	Output
. ALTNOM	ALTKM = ALTNOM	. ALTKM

Author: ____________ Date: 11/28/78

Diagram ID: 2.5.3 Name: CONDIA Description: DIAGRAM THE CONTROL NETWORK

Input

- samples/scene (NERSAM)
- NODT ('CON,' 'CHE,' 'ALL,' or 'ERR')
- numbers of nodes (NETHI)
- number of control points (NCTLPT)
- window number (NWNDOW)
- data check out mode (0 or 1) (MDATAC)
- scene information (NERTS)
- point numbers (NETPT)
- adjusted scanner coordinates for each node (ADJNET)
- WLIN (1)
- WSAM (2)

Process

1. Check for valid samples/scene.
2. Determine which points to diagram. If NODT = 'ERR', call DIAERR, (2.5.3.1) and then set KOMD to blanks and return.
3. Initialize.
4. Load diagram array with point numbers.
5. Diagram points above scene.
6. Diagram points within scene.
7. Diagram points below scene.

Output

- command from CONOOO (KOMD)
- control network diagram

Error messages

- error diagram

Author: ________

Date: 01/17/79

Diagram ID: 2.5.3.1

Name: DIAERR

Description: DIAGRAM CONTROL/CHECK POINT ERRORS

Input

. batch switch (MBATCH)

. type of error to be plotted (line or sample)

. sort parameter

Process

1. Determine if mode is batch or demand.
2. Set plot constants for demand mode, or
3. Set plot constants for batch mode.
4. Determine if line or sample error is to be plotted.
5. Set up sort according to sort parameter specified by user.
6. Check if network has been adjusted.
7. Begin sorting points.
8. Calculate and plot the errors.

Output

Error messages

. error diagram

Author:

Date: 11/28/79

Diagram ID: 2.5.4   Name: CONEXI   Description: TERMINATE CONTROL

Input	Process	Output
Registration Parameter File . number of fatal errors (NDFATL) Card image	1. If user has not requested that Registration Parameter File be saved, delete file. 2. If fatal errors, exit in error. 3. Exit.	

Author:

Date: 01/18/79

Diagram ID: 2.6 Name: PICTAB Description: DISPLAY LANDSAT TAPES ON LINE PRINTER

Input

Source data tape:
- Landsat

Registration Parameter File

Card images:
- User directives

Process

1. Initialize by opening files, loading registration parameters, identifying Landsat scene and queuing default commands (2.6.1).
2. Read commands and specifications.
3. If command defines or selects specifications, store specifications in applicable variables (2.6.2).
4. If command is DISPLAY, display the data for the current channel(s) within the current window by radiance or a function of the radiance (gradient, laplacian, class) (2.6.3).
5. If command is TABULATE, tabulate the number of pixels displayed by each symbol (2.6.4).
6. If command is FACTOR, generate the factor structure and factor coefficients from specified channels (2.7.5).

Output

Signature File
- Symbol lookup table
  - . type of data to process (raw, transform, polar)
  - . number of copies of output windows
  - . text headings
  - . origin coordinates for window
  - . radiance limits
  - . window scale
  - . sharpening filters
  - . window line increment
  - . window sample increment
  - . symbol table
  - . tick intervals
  - . vertices of windows
  - . linear transformation coefficients

Displays:
- Radiance
- Function of radiance (gradient, laplacian, class)

Tabulation
- . factor structure
- . factor coefficients

Author:

Diagram ID: 2.6 Name: PICTAB Date: 01/18/79

Description: DISPLAY LANDSAT TAPES ON LINE PRINTER

Input	Process	Output
	7. If command is ROTATE, rotate the structure and coefficients generated by FACTOR command (2.7.6).	
	8. If command is PARTITION, control the partitioning of the data as specified: DENSITY, GRADIENT, or LAPLACIAN (2.7.7).	Transformation Representation Matrix File (mean probability Surface) Signature File - Label lookup table - Symbol lookup table
	9. If command is EXIT, close files and exit from program (2.6.10).	
	10. If command is invalid, print an error message.	Error message
	11. Repeat steps 2 through 10 until exit from program.	

Author: ______

Diagram ID: 2.6.1 Name: PICXQT

Date: 01/18/79

Description: OPEN FILES AND IDENTIFY LANDSAT SCENE

Input	Process	Output
Source data tape - Landsat Registration Parameter File	1. Open MSS data file (tape), read header and ID records, and store information. 2. Print header information to identify tape. 3. Load registration parameters from Registration Parameter File; otherwise, use nominal (default) registration parameters. 4. Queue default commands.	. processing flags . data record length . Landsat project number . days since launch . GMT . mode correction code . date . sun elevation angle . number of channels . number of lines in full scene . number of samples in full scene . scene center . CCT number Identification of Landsat tape . coordinate transformation . coefficients for converting from one coordinate system to another

Author: ____________ Date: ____________

Diagram ID: 2.6.2 Name: ____________ Description: PROCESS COMMANDS THAT SPECIFY AND STORE VARIABLES

Input

. command word

Card image

Process

1. If command is ALIGN, get points to align and recompute scene center (2.7.2.1).
2. If command is CHANNEL, get channel number and type of channel to display or process (2.7.2.2.).
3. If command is SYMBOL, get information for symbol table (2.6.2.3).
4. If command is LINEAR, get transformation coefficients to be used in subsequent processing of linear channels (2.7.2.4).
5. If command is ORIGIN, get coordinate system and coordinates of the origin of the scene (2.7.2.5).
6. If command is POLAR, get gains and biases to be used in subsequent processing of polar channels (2.7.2.11).
7. If command is RADIANCE, get radiance or range for channel specified by CHANNEL command (2.7.2.6).

Output

. scene center line

. scene center sample

. transformation coefficients

. channels to display

. type of channel to display

. maximum number of channels

. symbol table

. linear gain

. linear weight

. weighted gain

. system name of coordinates

. window packet

. polar gain

. polar bias

. minimum radiance

. maximum radiance

Author: ______ Date: ______

Diagram ID: 2.6.2 Name: ______ Description: PROCESS COMMANDS THAT SPECIFY AND STORE VARIABLES

Input	Process	Output
	8. If command is SHARPENING, get the coefficients for a sharpening filter (2.7.2.12)	. sharpening coefficients
	9. If command is SPACING, get the MSS line and sample increments to be used for processing subsequent windows (2.7.2.7).	. line increment . sample increment
	10. If command is TICK, get the coordinate system and the intervals for primary and secondary ticks (2.7.2.8).	. tick system name . primary tick interval . secondary tick interval
	11. If command is WINDOW, get the coordinate system and the coordinates of that system for the vertices of a window (2.7.2.9).	. vertex system name . vertices
	12. If command is ZONE, get the universal transverse mercator (UTM) zone number and calculate UTM central meridian (2.7.2.10).	. UTM central meridian . UTM zone number

Author:

Diagram ID: 2.6.2.3 Name: KMDSYM Date: Description: GET/CHECK WINDOW SYMBOLS

Input

Card image

. confirm switch (MCFIRM)

Process

1. If no data field is present then print symbol legend and branch to step 12.
2. Save total counter of diagnostics.
3. Get symbol and print a message if it is *, or + which are invalid.
4. Get minimum number value for symbol.
5. Initialize for implicit maximum symbol and number by using the value, from steps 3 and 4.
6. Pick up and check explicit maximum symbol, same as step 3.
7. If number present, store as maximum number.
8. Compute symbols per number.
9. If any diagnostics occurred, branch to step 12.
10. Fill the symbol table.
11. If confirm switch is on, print confirmation.
12. Blank command word.

Output

Error message

. symbol table (KSYM)

Symbol table

. command name

Author:

Date: 01/30/79

Diagram ID: 2.6.3 Name: PICDIS Description:

Input

Card image

. symbol table

. data checkout flag (MDATAC)

. name of coordinate system (KSYOWW)

. output window packet (MSAOWW)

. input window packet (MSAIWW)

Process

1. Get type of display and save (default is RADIANCE).
2. If type of display is invalid, print an error message and set an error flag.
3. If any other information on card, print an error message and set an error flag.
4. If symbol table has not been set, print a message and set an error flag.
5. If data checkout flag is on, skip to step 15.
6. To prepare for output:
   a. Determine if there is to be overprinting.
   b. Compute printer/plotter spacing.
   c. Compute the envelopes in needed coordinates.
7. Crop output window to fit input window.

Output

. type of display

Error message

. error flag

. output window packets

Window packet information:

. adjusted scanner (MSAOWW)

. geographic (GEDOWW)

. UTM (UTMOWW)

. printer/plotter (PPDOWW)

Author:

Date: 01/30/79

Diagram ID: 2.6.3 Name: PICDIS Description:

Input	Process	Output
	8. Initialize page number and window number.	. window number (NWNDOW) . page number
	9. Generate tick intervals and save.	. tick table
	10. Print window headings.	Window headings
	11. If type of display is RADIANCE, display by radiance (2.6.3.1).	Displays: - Radiance
	12. If type of display is GRADIENT, display gradient (2.6.3.2).	- Gradient
	13. If type of display is CLASS, display class (2.6.3.3).	- Class
	14. If type of display is LAPLACIAN , display Laplacian (2.6.3.4).	- Laplacian
	15. Blank command name	. command name (KOMD)

Author:

Diagram ID: 2.6.3.1.8 Name: GETRAD Date: 01/18/79

Description: GET ORIGINAL/TRANSFORMED RADIANCE FROM LANDSAT TAPE

Input

- channels to display
- number of channels

Source data tape
  Landsat

- sharpening coefficients
- type of channel
- weighted gains coefficients

Process

1. Read and unpack original Landsat channel(s).
2. If there are sharpening filter coefficients, do the following for specified channels:
   a. Save radiances that are in buffer.
   b. Determine slope state (negative or positive).
   c. If slope state changes from
      (1) Positive to negative, sharpen up.
      (2) Negative to positive, sharpen down.
3. Set up buffer pointers for transformed channels.
4. If there are transformations to do, for each value unpacked:
   a. Do linear transformations.
   b. If there are polar transformations, do them.

Output

- unpacked radiance buffers (IRADBF)

Author: ______

Diagram ID: 2.6.3.1 Name: PICDI3 Description: DISPLAY RADIANCE Date: ______

Input

- window packet (MSAOWW)

Source data tape
- Landsat

- type of channel
- channel(s) to display

Process

1. Compute size of print window:

   n = number of pixels (columns)/line

   X = number of lines in window

   number of pixels = n*X

2. Initialize read window and position tape at top of window.

3. Zero local frequency table.

4. Generate and print sample scale and border.

5. Initialize output buffer to no data symbols.

6. Read, unpack, and, if applicable, transform the data for n pixels in all channels (2.6.3.1.8).

7. If current pixel is not included in output window, ignore and go to step 13.

Output

Display:
- Sample scale and border

- output buffer

Author: ____

Date: ____

Diagram ID: 2.6.3.1 Name: PICDI3 Description: DISPLAY RADIANCE

Input	Process	Output
. channel(s) to display . radiance ranges/channel . symbol table	8. If radiance value of pixel in first channel does not fall within the specified valid range, store a no data symbol in output buffer; skip to step 13. 9. If radiance value of pixel in second channel does not fall within the specified valid range, store a no data symbol in output buffer and go to the next pixel in first channel (step 8). 10. Repeat step 8 for all specified channels. 11. Using radiance value from first channel, pick up symbol from the symbol table and store in output buffer. 12. Add 1 to count of frequency of this radiance value from first channel. 13. Repeat steps 6 through 11 for n pixels. 14. Print output buffer. 15. Repeat steps 5 through 14 for all groups of n pixels until all pixels are processed.	. output buffer Display: - Data (output buffer)

Author:

Date:

Diagram ID: 2.6.3.1 Name: PICDI3 Description: DISPLAY RADIANCE

Input	Process	Output
	16. Print border and sample scale. 17. Move data from local frequency tables into global table.	Display: - Border and sample scale . frequency table

Autnor:

Date: 02/05/79

Diagram ID: 2.6.3.2 Name: PICDI4 Description: DISPLAY GRADIENT

Input

. window packet (MSAOWW)

Source data tape:
Landsat

Transformation Matrix File

. type of channel (raw, transform, etc.)

Process

1. Compute size of print window:

   n = number of pixels (columns)/line

   X = number of lines in window

   number of pixels = n*X

2. Initialize read window and position tape at top of window.

3. Zero local frequency table.

4. If gradient transformation matrix for this window is not in file, compute gradient and store transformation matrix (2.6.7.12).

5. Reposition tape at top of window.

6. Generate sample, scale, and border and print.

7. Initialize output buffer to no data symbols.

8. Read, unpack, and, if applicable, transform the data for n pixels in all channels (2.6.3.1.8).

Output

. transformation matrix

Display:
- Sample scale and border

Author: ________ Date: 02/05/79

Diagram ID: 2.6.3.2 Name: PICDI4 Description: DISPLAY GRADIENT

Input

. channel(s) to display

. symbol table

Process

9. If current pixel is not included in output window, ignore and go to step 13.
10. Using a pixel's value from each of the 2 specified channels as indices to the transformation matrix, pick up gradient value (GRAD) at that location.
11. Using value GRAD, look up symbol in the symbol table and store in output buffer.
12. Add 1 to count of frequency of this gradient value.
13. Repeat steps 9 through 10 for n pixels.
14. Print output buffer.
15. Repeat steps 7 through 14 for all groups of n pixels until all pixels have been processed.
16. Print border and sample scale.
17. Move data from local frequency table to global table.

Output

. output buffer

Display:

- Data (output buffer)
- Border and sample scale

. frequency table

Author: ______

Diagram ID: 2.6.3.3 Name: PICDI5 Date: 01/30/79

Description: DISPLAY CLASS

Input	Process	Output
. window packet	1. Compute size of print window: n = number of pixels (columns)/line X = number of lines in window number of pixels = n*X	
. label lookup table	2. If there is no label lookup table, print a message and return.	Error message
Source data tape: Landsat	3. Initialize read window and position tape at top of window.	
	4. Zero local frequency table.	
	5. Generate and print sample scale and border.	Display: - Sample scale and border
	6. Initialize output buffer to no data symbols.	
. type of channel (raw, transform, etc.)	7. Read, unpack, and, if applicable, transform data for n pixels (2.6.3.1.8).	
	8. If current pixel is not included in output window, ignore and go to step 15.	

Author:

Diagram ID: 2.6.3.3 Name: PICDIS Description: DISPLAY CLASS Date: 01/30/79

Input

. channel(s) to display

. symbol table

Process

9. Using a pixel's value from each of 2 specified spec- fied channels (F_1 and F_2) as indices to the label lookup table, determine the class of the pixel.
10. Use the class of the pixel as value to apply to symbol table to determine the symbol.
11. Store the symbol in output buffer.
12. Add 1 to count of frequency of this class value.
13. Repeat 7 through 12 for n pixels.
14. Print output buffer.
15. Repeat 6 through 14 for all groups of n pixels.
16. Print border and sample scale.
17. Move data from local frequency table into global table.

Output

. output buffer

Display:

- Data (output buffer)
- Border and sample scale

. frequency table

Author: ____________ Date: ____________

Diagram ID: 2.6.3.4 Name: PICDI6 Description: DISPLAY LAPLACIAN

Input	Process	Output
. window packet	1. Compute size of print window: n = number of pixels (columns)/line X = number of lines in window number of pixels = n*X	
Source data tape - Landsat	2. Initialize read window and position tape at top of window.	
	3. Zero local frequency table.	
Transformation Matrix File	4. If Laplacian transformation matrix for this window is not in file, compute Laplacian and store transformation matrix (2.7.7.13).	. transformation matrix
	5. Reposition tape at top of window.	
	6. Generate and print sample scale and border.	Display: - Sample scale and border
	7. Initialize output buffer to no data symbols.	
. type of channel . number of channels	8. Read, unpack, and, if applicable, transform data for n pixels (2.6.3.1.8).	

Author: ____

Date: ____

Diagram ID: 2.6.3.4 Name: PICDI6 Description: DISPLAY LAPLACIAN

Input

. channel(s) to display

Process

9. If current pixel is not included in output window, ignore and go to step 13.
10. Using a pixel's value from each of 2 specified channels as indices to the transformation matrix, pick up Laplacian value (LAPL) at that location.
11. Using value LAPL, look up symbol in the symbol table and store in output buffer.
12. Add 1 to count of frequency of this Laplacian value.
13. Repeat steps 9 through 12 for n pixels.
14. Print output buffer.
15. Repeat steps 7 through 13 for all groups of n pixels.
16. Print border and sample scale.
17. Move data from local frequency table to global table.

Output

. output buffer

Display:

- Data (output buffer)

. frequency table

Author:

Date: 03/08/79

Diagram ID: 2.6.10 Name: PICEXI Description: EXIT FROM PICTAB

Input	Process	Output
. data/checkout flag Source data tape: - Landsat . fatal error flag	1. If in data checkout mode, exit. 2. Close Landsat tape file. 3. If fatal errors occurred and user does not want displays printed, delete any print files. 4. If fatal errors did not occur, or fatal errors occurred and user wants displays printed anyway, close print files. 5. Exit.	

Author:

Diagram ID: 2.7 Name: CLRTAB

Date: 01/18/79

Description: DISPLAY LANDSAT TAPES ON CRT

Input	Process	Output
Source data tape - Landsat Registration Parameter File Card images - User directives	1. Initialize by opening files, loading registration parameters, identifying Landsat scene, and queuing default commands (2.7.1). 2. Read commands and specifications. 3. If command defines or selects specifications, store specifications in applicable variables (2.7.2). 4. If command is DISPLAY, display the data for the current channel(s) within the current window by radiance or a function of the radiance (gradient, laplacian, class) (2.7.3). 5. If command is TABULATE, tabulate the number of pixels displayed by each symbol (2.7.4). 6. If command is FACTOR, generate the factor structure and factor coefficients from specified channels (2.7.5).	Signature File - Symbol lookup table . type of data to process (raw, transform, polar) . number of copies of output windows . text headings . origin coordinates for window . radiance limits . window scale . sharpening filters . window line increment . window sample increment . color symbol table . tick intervals . vertices of windows . linear transformation coefficients Displays - Radiance - Function of radiance (gradient, laplacian, class) Tabulation . factor structure . factor coefficients

Author:

Diagram ID: 2.7 Name: CLRTAB Description: DISPLAY LANDSAT TAPES ON CRT

Date: 01/18/79

Input	Process	Output
	7. If command is ROTATE, rotate the structure and coefficients generated by the FACTOR command (2.7.6).	. rotated factor structure . rotated factor coefficients
	8. If command is PARTITION, control the partitioning of the data as specified: DENSITY, GRADIENT, or LAPLACIAN (2.7.7).	Transformation Representation Matrix File (mean probability surface) Signature File - Label lookup table - Symbol lookup table
	9. If command is HISTOGRAM, display a histogram (frequency distribution) of radiance or class (2.7.8).	Displays - Histogram
	10. If command is PROFILE, plot the radiance values of consecutive pixels along a specified line (perhaps irregular) without destroying background color (2.7.9).	- Profile
	11. If command is EXIT, close files and exit from program (2.7.10).	
	12. If command is invalid, print an error message.	
	13. Repeat steps 2 through 12 until exit from program.	

Author: ______ Date: 01/18/79

Diagram ID: 2.7.1 Name: CLRXQT Description: OPEN FILES AND IDENTIFY LANDSAT SCENE

Input	Process	Output
Source data tape - Landsat Registration Parameter File	1. Open MSS data file (tape), read header and ID records, and store information. 2. Print header information to identify tape. 3. Load registration parameters from registration parameter file. If not available, load nominal registration parameters. 4. Queue default commands.	. processing flags . data record length . Landsat project number . days since launch . GMT . mode correction code . date . sun elevation angle . number of channels . number of lines in full scene . number of samples in full scene . scene center . CCT number Identification of Landsat tape . registration parameters (coordinate transformation coefficients for converting from one coordinate system to another)

Author: ______ Date: ______

Diagram ID: 2.7.2 Name: ______ Description: COMMANDS THAT SPECIFY AND STORE VARIABLES

Input	Process	Output
. command word Card image	1. If command is ALIGN, get points to align and recompute scene center (2.7.2.1).	. scene center line . scene center sample . coefficients for corrected STM
	2. If command is CHANNEL, get channel number and type of channel to display or process (2.7.2.2).	. channels to display . type of channel to display . maximum number of channels
	3. If command is COLOR, get information for color symbol table (2.7.2.3).	. initial color symbol table
	4. If command is LINEAR, get transformation coefficients to be used in subsequent processing of linear channels (2.7.2.4).	. linear gain . linear weight . weighted gain
	5. If command is ORIGIN, get coordinate system and coordinates of the origin of the window (2.7.2.5).	. window packet coordinates
	6. If command is RADIANCE, get radiance limits or range for channel specified by CHANNEL command (2.7.2.6).	. minimum radiance . maximum radiance
	7. If command is SPACING, get the MSS line and sample increments to be used for processing subsequent windows (2.7.2.7).	. line increment . sample increment

Author: ______ Date: ______

Diagram ID: 2.7.2 Name: ______ Description: COMMANDS THAT SPECIFY AND STORE VARIABLES

Input	Process	Output
	8. If command is TICK, get the coordinate system and the intervals for primary and secondary ticks (2.7.2.8).	. tick system . primary tick interval . secondary tick interval
	9. If command is WINDOW, get the coordinate system and coordinates of the vertices of a window (2.7.2.9).	. coordinate vertex system . vertices
	10. If command is ZONE, get the universal transverse mercator (UTM) zone number and calculate UTM central meridian (2.7.2.10).	. UTM zone number . UTM central meridian
	11. If command is POLAR, get gains and biases to be used in subsequent processing of polar channels (2.7.2.11).	. polar gain . polar bias

Author: ____________ Date: ____________

Diagram ID: 2.7.2.1 Name: KMDALI Description: DETERMINE NEW TRANSFORMATION COEFFICIENTS FOR REALIGNMENT

Input

Card image

Process

1. Get scanner (line length adjusted) coordinates.
2. Correct scanner coordinates.
3. Pick up name of coordinate system to be used. → . window packet (KSYOWW)
4. If coordinate system is DEGREES (geographic):
   a. Pick up and check validity of table.
   b. Pick up and check validity of command.
   c. If the difference between the location of the central meridian and the input longitude is 74, set the central meridian to current longitude. → . central meridian (STMCMD)
5. If coordinate system is kilometers or meters:
   a. If needed, convert from kilometers to meters.
   b. If UTM zone is not specified, print a message. → Warning message

Output

. window packet (KSYOWW)

. central meridian (STMCMD)

Warning message

Author: ____________ Date: ____________

Diagram ID: 2.7.2.1 Name: KMDALI Description: DETERMINE NEW TRANSFORMATION COEFFICIENTS FOR REALIGNMENT

Input	Process	Output
	c. Pick up and check validity of easting.	
	d. Pick up and check validity of northing.	
	e. Convert from UTM to geographic coordinates.	
	6. Compute the deltas between the old alignment in STM coordinates.	
	7. Use STM coordinates to modify transformation coefficients.	. transformation coefficients for STM (CORSTM)
	8. Revert bilinear transformation coefficients.	. transformation coefficients for corrected STM (STMCOR)
	9. Confirm coordinates (geographic and UTM).	Coordinates
	10. Mark origin as destroyed.	. window packet (KSYOWW)
	11. Recompute scene center.	. scene line center (CTRLIN)
		. scene center sample (CTRSAM)
	12. Blank command word.	. command word (KOMD)

Author: ____________ Date: ________

Diagram ID: 2.7.2.2 Name: KMDCHA Description: SPECIFIES CHANNELS FOR PROCESSING

Input	Process	Output
Card image	1. If no specification field with command, branch to step 9.	
	2. Save diagnostic count.	. diagnostic count (NDTOTL)
	3. Delete old transform type, channels, and radiance limits (set to zero).	. transform type . radiance range
	4. Pick up channel type (default is RAW). If invalid channel type, print message and set error flag.	Error message . error flag
. number of channels (NERCHA)	5. Set maximum number of channels: a. Set to 2 if type of channel is LINEAR or POLAR. b. Set to NERCHA if channel is RAW.	. maximum number of channels (NLCMAX)
	6. Get explicit channel numbers. If invalid or duplicate channels, print message and set error flag.	. channel numbers (LIMCH)
	7. If errors occurred, reinitialize and go to step 10.	

Author:

Date:

Diagram ID: 2.7.2.2 Name: KMDCHA Description: SPECIFIES CHANNELS FOR PROCESSING

Input	Process	Output
	8. Assign buffers for implicit RAW channels.	
	9. If confirmation switch is on, print confirmation information.	Channel type Channel numbers
	10. Blank command word.	. command word (KOMD)

Author:

Date: 03/07/79

Diagram ID: 2.7.2.3 Name: KMDCOL Description: DEFINES INITIAL COLOR SYMBOL TABLE

Input

Card image

. initial color symbol table

. color symbol table

Process

1. If no specifications on card image:

   a. Print initial color symbol table.

   b. If color symbol table is defined, print it.

   c. Go to step 14.

2. If first COLOR command, blank initial color symbol table and initialize color pointer to zero.

3. Get first color name. If invalid, print message and set error flag.

4. Get low value associated with first color name.

5. If invalid, print message and set error flag.

6. Get second color name.

7. If not present or invalid, print message and set error flag.

8. Get high value associated with second color name.

9. If invalid, print message and set error flag.

Output

Initial color symbol table

Color symbol table

Error message

. error flag

Author:

Diagram ID: 2.7.2.3 Name: KMDCOL

Date: 03/07/79

Description: DEFINES INITIAL COLOR SYMBOL TABLE

Input	Process	Output
	10. If any duplicates of color pairs and values, print a message and set error flag. 11. If any errors occurred, go to step 14. 12. Add 1 to color pointer. 13. Store the two colors and associated values in initial color table. 14. Blank command word.	. initial color symbol table . command word

Author: ____________ Date: ____________

Diagram ID: 2.7.2.4 Name: KMDLIN Description: GET/CHECK LINEAR WEIGHTS AND GAINS

Input

. current linear bias (LRTB12)

Card image

. number of channels (NERCHA)

. confirmation switch (MCFIRM)

Process

1. Retrieve current biases.
2. Get linear transformed channel number.
3. If invalid channel, print message and set error flag.
4. Get transformation coefficient type (WEIGHTS, GAIN, or BIAS).
5. If coefficient type is invalid, print a message and set error flag and go to step 9.
6. Undate weights or gains if either GAIN or WEIGHT.
7. Update weighted gains.
8. If need confirmation, print confirmation.
9. Continue processing until no other specifications.

Output

Error message

. error flag

linear gain (RTLGAN)

. linear weight (RTLWGT)

. weighted gain (LRTW12)

Gains

Weights

Weighted gain

Author:

Date:

Diagram ID: 2.7.2.5

Name: KMDORI

Description: GETS/CHECKS LOCATION OF ORIGIN OF WINDOW (DISPLAY)

Input	Process	Output
Card image . coordinate system (KSYOWW) . confirmation switch (MCFIRM)	1. Check for valid coordinate system of SCA, DEG, KM, MET (use old system if none specified). If invalid, print error message and set error flag. 2. Pick up and save coordinates in the coordinate system specified. 3. Convert coordinates to the other three coordinate systems not input (A2G, U2G, G2A, G2U, P2A). 4. If confirm switch is on, print confirmation information. 5. Save type of coordinate system specified. 6. Blank command word.	Error message . error flag Window packet coordinates: . adjusted scanner (MSAOWW) . geographic (GEDOWW) . UTM (UTMOWW) . printer/plotter (PPDOWW) Coordinate information . coordinate information for origin (KSYOWW) . command word (KOMD)

Author: ____________ Date: ____________

Diagram ID: 2.7.2.6 Name: KMDRAD Description: GET/CHECK RADIANCE LIMITS

Input

Card image

. number of channels (NERCHA)

. channels to process

Process

1. If no specification field on card image, branch to step 3.
2. If no CHANNEL command processed before this RADIANCE command, skip picking up radiance information.
3. If already processed CHANNEL command, then for each channel:
   a. Get minimum radiance value. If invalid, print error message and set error flag.
   b. Get maximum radiance value. If invalid, print error message and set error flag.
4. If confirmation switch is on, print confirmation information.
5. Blank command word.

NOTE: Must have already had CHANNEL command.

Output

Error message

. error flag

. minimum radiance (LCVLO)

. maximum radiance (LCVHI)

Radiance ranges

. command word (KOMD)

Author: ______

Date: ______

Diagram ID: 2.7.2.7

Name: KMDSPA

Description: SPECIFIES LINE AND SAMPLE INCREMENTS

Input	Process	Output
Card image . confirmation switch (MDATAC)	1. Get and save window spacing information. If invalid, print error message and set error flag. 2. If confirmation switch is on, print confirmation. 3. Blank command word.	. window packet array . line spacing . sample spacing Error message . error flag Window spacing information . command word (KOMD)

Author: ______

Diagram ID: 2.7.2.8 Name: KMDTIC Date: ______

Description: GET/CHECK WINDOW TICK INTERVALS

Input	Process	Output
Card image . coordinate system (KSYOWW) . confirmation switch (MCFIRM)	1. Initialize to allow input for this command to span 2 cards (SPANS). 2. Get coordinate system (default is old coordinate system). a. If bad coordinate system, go to step 10. 3. If coordinate system is SCAN: a. Pick up and check line and sample location of primary tick. b. If confirm switch is on, print confirmation. c. Go to step 8. 4. If system is DEG: a. Set maximum number of sexagenarian places. b. Pick up and check latitude and longitude. c. If confirm switch is on, write confirmation. d. Go to step 8.	. coordinate system (KSYOWW) . window packet (GEDOWW) Tick intervals

Author: ______

Diagram ID: 2.7.2.8 Name: KMDTIC Date: ______

Description: GET/CHECK WINDOW TICK INTERVALS

Input	Process	Output
	5. If system is MIN: a. Set conversion factor to convert from minutes to degrees. b. Pick up and check latitude and longitude and convert to degrees. c. Save as minutes. d. If confirm switch is on, print confirmation. e. Go to step 8. 6. If system is KM: a. Set conversion factor to convert to meters. b. Pick up, check, and convert kilometers (UTM coordinates). c. If confirm switch is on, print confirmation. d. Go to step 8. 7. If system is MET: a. Pick up and check UTM coordinates.	Tick intervals

Author: ______ Date: ______

Diagram ID: 2.7.2.8 Name: KMDTIC Description: GET/CHECK WINDOW TICK INTERVALS

Input	Process	Output
	b. If confirm switch is on, print confirmation. 8. Store name of coordinate system. 9. If secondary tick information has not been processed, then repeat steps 2 through 9 using the coordinate system of primary ticks if none is specified for secondary. 10. Blank command word.	Tick interval . window packet (KSYOWW) . command word (KOMD)

Author: ______ Date: ______

Diagram ID: 2.7.2.9 Name: KMDWIN Description: GETS/CHECKS VERTICES OF WINDOW

Input	Process	Output
Card image . coordinate information (KSYOWW) . confirmation switch (MCFIRM)	1. Initialize for multiple input cards. 2. Get coordinate system specified (use old coordinate system if none specified). If invalid, print message and set error flag. 3. Depending on the coordinate system: a. Determine if polygon needs to be closed — close when no more vertices. b. Get coordinates for a vertex. If invalid, print message and set error flag. c. Save vertex values in appropriate array. d. Repeat steps a through c until no more vertices are supplied. 4. If confirm switch is on, print confirmation information. 5. Blank command word.	Error message . error flag . coordinate system for vertices (KSYOWW) Window packet information: . adjusted scanner (MSAOWW) . geographic (GEDOWW) . UTM (UTMOWW) . printer/plotter (PPDOWW) Location of vertices . command word (KOMD)

Author: ____

Date: ____

Diagram ID: 2.7.2.10

Name: KMDZON

Description: GET/CHECK UTM PROJECTION ZONE

Input	Process	Output
Card image . confirmation switch (MCFIRM)	1. If UTM central meridian is not yet set, set to 30,000. 2. Calculate (183.-UTMCMD)/6. 3. Retrieve the zone number. 4. Calculate 183-6*ITEMP. 5. If confirm switch is on, print confirmation. 6. Flag the origin as destroyed. 7. Clear extra zone specifications. 8. Blank command word.	. output window packet mode names (KSYOWW) . UTM zone (UTMCMD) . UTM central meridian Zone Central Meridian . command word (KOMD)

Author: 　　　　　　　　Date:

Diagram ID: 2.7.2.12　　Name: KMDSHA　　Description: PICK UP FILTER COEFFICIENTS FOR SHARPENING

Input	Process	Output
Card image . number of channels (NERCHA) . confirmation switch (MCFIRM)	1. Save current coefficients. 2. Get channel number to be sharpened. If invalid, print message and set error flag. 3. Get axis to be sharpened (can only be SAMPLE axis). If invalid, print message and set error flag. 4. Get coefficients. If invalid, print message and set error flag. 5. If confirm switch is on, print confirmation. 6. Blank command word.	Error message . error flag . sharpening filter coefficients (IRSF12) Channel and coefficients . command word (KOMD)

Author:

Diagram ID: 2.7.2.13 Name: KMDINT

Date: 03/07/79

Description: DEFINES INITIAL INTENSITY TABLE

Input

Card image

. initial intensity table

. intensity table

Process

1. If no specifications on card image:

   a. Print initial intensity table.

   b. If intensity table exists, print it.

2. If first INTENSITY command, blank initial intensity table and set entry pointer to zero.

3. Get first scaling factor.

4. If invalid, print message and set error flag.

5. Get low value associated with first scaling factor.

6. If invalid, print message, set error flag.

7. Get second scaling factor.

8. If invalid, print message and set error flag.

9. Get high value associated with second scaling factor.

10. If invalid, print message and set error flag.

Output

Initial intensity table

Intensity table

Error message

. error flag

Author: ____________ Date: 03/07/79

Diagram ID: 2.7.2.13 Name: KMDINT Description: DEFINES INITIAL INTENSITY TABLE

Input	Process	Output
	11. If any errors occurred, go to step 14. 12. Add 1 to pointer to intensity table. 13. Save the two scaling factors and associated values in intensity table. → 14. Blank command word. →	. initial intensity table . command word

Author: ______

Date: 02/16/79

Diagram ID: 2.7.3 Name: CLRDIS Description: PREPARE FOR DISPLAYS

Input	Process	Output
Card image	1. Get type of display and save.	. type of display
	2. If invalid type, print message; save default of RADIANCE.	Error messages
	3. If anything else on card, print an error message.	
. data checkout flag (MDATAC)	4. If in checkout mode, return.	
	5. To prepare for output, compute printer/plotter spacing and scaling and compute the envelopes in needed coordinates.	. output window packets
. input window packet . output window packet	6. Crop output window to fit input window.	
	7. Initialize display number and window number. Calculate size of display window.	. display number . window number
. input window packets . scene ID	8. Generate display header buffer with scene information and ancillary data such as scaling information needed for display.	. header buffer
	9. Generate all ticks.	. tick table
	10. Set up color and intensity tables; set up RAMTEK lookup table (2.7.3.6).	

Author:

Diagram ID: 2.7.3 Name: CLRDIS Date: 02/16/79

Description: PREPARE FOR DISPLAYS

Input

. type of channel

Process

11. Determine if spatial content of data is changed (2.7.3.7).
12. If type of channel is POLAR, do a display with varying color intensities (2.7.3.5).
13. If type of display is RADIANCE, display radiance (2.7.3.1).
14. If type of display is GRADIENT, display gradient (2.7.3.2).
15. If type of display is CLASS, display class (2.7.3.3).
16. If type of display is LAPLACIAN, display Laplacian (2.7.3.4).
17. Output buffers, if applicable, to color graphics.

Output

. header record
. RAMTEK refresh memory buffer
. tick buffer

Display:
- Radiance
  Varying color intensity
  Multicolor display
- Gradient
- Class
- Laplacian

Author: ____________ Date: 02/05/79

Diagram ID: 2.7.3.1 Name: CLRDI3 Description: DISPLAY RADIANCE

Input	Process	Output
. window packet (MSAOWK)	1. If output window is > 512 by 512 pixels, crop to that size.*	
Source data tape - Landsat	2. Initialize read window and position tape at top of window (according to sample).	
	3. Zero local frequency table.	
. type of channel	4. If type of channel is not RAW or POLAR, print error, set error flag, and return.	
. number of channels (Source data tape)	5. Read, unpack, and, if applicable, transform the data for n pixels in all channels (2.6.3.1.8).	
. tick table	6. If tick should be output at current pixel location, store tick.	. output tick buffer
	7. If current pixel is not included in output window, go to step 12.	
. channel to display	8. If radiance value of pixel in current channel does not fall within the specified valid range, ignore and go to next pixel.	

*Determine: n = number of pixels (columns)/line
groups of n pixels = number of lines in window

Author:

Date: 02/05/79

Diagram ID: 2.7.3.1 Name: CLRDI3 Description: DISPLAY RADIANCE

Input	Process	Output
. radiance ranges for channels . color symbol table intensity table	9. Repeat step 8 for all specified channels. 10. If type is RAW, use radiance value from the channel, pick up color and brightness values from color table and intensity table and store in RAMTEK refresh memory buffer. 11. If type of channel is POLAR use the color and intensity table to assign a value to the current pixel based on brightness (polar channel 1) and color angle (polar channel 2) and store value in output buffer. 12. Add 1 to count of frequency of this radiance value from first channel. 13. Repeat steps 6 through 12 for n pixels until all pixels are processed. 14. Repeat steps 5 through 13 for all groups of n pixels until all pixels are processed. 15. Move data from local frequency tables into global table.	. RAMTEK refresh memory buffer . frequency table

Author:

Date: 02/05/79

Diagram ID: 2.7.3.1 | Name: CLRDI3 | Description: DISPLAY RADIANCE

Input

. type of display
. window packet
. type of channel

Process

16. Save current display type, window packet, and type of channel for later checking.

Output

. previous type of display
. previous window packet
. previous type of channel

Author:

Date: 02/05/79

Diagram ID: 2.7.3.2 Name: CLRDI4 Description: DISPLAY GRADIENT

Input	Process	Output
. window packet (MSAOWW)	1. If output window is > 512 by 512 pixels, crop to that size.*	
Source data tape - Landsat	2. Initialize read window and position tape at top of window.	
	3. Zero local frequency table.	
Transformation Matrix File	4. If gradient transformation matrix for this window is not in file, compute gradient and store transformation matrix (2.6.7.12).	. transformation matrix
	5. Reposition tape at top of window, if necessary.	
. type of channel (raw, transform, etc.)	6. Read, unpack, and, if applicable, transform data for n pixels in all channels (2.6.3.1.8).	
	7. If a tick should be output at the current pixel location, store tick information.	. output tick buffer
	8. If current pixel is not included in output window, go to step 12.	

*Determine: n = number of pixels (columns)/line
groups of n pixels = number of lines in window

Author:

Date: 02/05/79

Diagram ID: 2.7.3.2 Name: CLRDI4 Description: DISPLAY GRADIENT

Input	Process	Output
. channel(s) to display	9. Using a pixel's value from each of the 2 specified channels as indices to the transformation matrix, pick up gradient value (GRAD) at that location.	
. color table . intensity table	10. Using value GRAD, determine the color and brightness from the color and intensity tables and store in RAMTEK refresh memory buffer in same relative pixel location as the input pixel.	. RAMTEK refresh memory buffer
	11. Add 1 to count of frequency of this gradient value.	
	12. Repeat steps 7 through 11 for n pixels.	
	13. Repeat steps 6 through 12 for all groups of n pixels until all pixels have been processed.	
	14. Move data from local frequency table to common table.	. frequency table
. type of display . window packet . type of channel	15. Save current display type, window packet and channel type for later checking.	. previous type of display . previous window packet . previous type of channel

Author: ______

Diagram ID: 2.7.3.3 Name: CLRDI5 Date: 01/30/79 Description: DISPLAY CLASS

Input

. window packet

. label lookup table

Source data tape

- Landsat

. type of channel (raw, transform)

. channel(s) to display

. color table

. intensity table

Process

1. If output window is > 512 by 512 pixels, crop to that size.*
2. If there is no label lookup table, print a message and return.
3. Initialize read window and position tape at top of window.
4. Zero local frequency table.
5. Read, unpack, and, if applicable, transform data for n pixels (2.6.3.1.8).
6. If a tick should be output at a current pixel location, store tick information.
7. If current pixel is not included in output window, go to step 12.
8. Using a pixel's value from each of two specified channels (F_1 and F_2) as indices to the label look-up table, determine the class of the pixel.
9. Use the class of the pixel as value to apply to color table and intensity table to determine color and brightness.

Output

Error message

. output tick buffer

*Determine: n = number of pixels (columns)/line
groups of n pixels = number of lines in window.

Author:

Date: 01/30/79

Diagram ID: 2.7.3.3 Name: CLRDI5 Description: DISPLAY CLASS

Input

. type of display
. window packet
. type of channel

Process

10. Store the color and brightness in RAMTEK refresh memory buffer.
11. Add 1 to count of frequency of this class value from the label lookup table.
12. Repeat steps 6 through 11 for n pixels.
13. Repeat steps 5 through 12 for all groups of n pixels.
14. Move data from local frequency table into global table.
15. Save current display type, window packet, and channel type for later checking.

Output

. RAMTEK refresh memory buffer
. frequency table
. previous type of display
. previous window packet
. previous type of channel

Author: ______ Date: ______

Diagram ID: 2.7.3.4 Name: CLRDI6 Description: DISPLAY LAPLACIAN

Input	Process	Output
. window packet (MSAOWW)	1. If output window is > 512 by 512 pixels, crop to that size.*	
Source data tape - Landsat	2. Initialize read window and position tape at top of window.	
	3. Zero local frequency table.	
Transformation Matrix File	4. If Laplacian transformation matrix for this window is not in file, compute Laplacian and store transformation matrix (2.6.7.13).	. transformation matrix
	5. Reposition tape at top of window, if necessary.	
. type of channel	6. Read, unpack, and, if applicable, transform data for n pixels (2.6.3.1.8).	
. tick table	7. If a tick should be output at the current pixel location, store information.	. output tick buffer.
	8. If current pixel is not in output window, go to step 12.	
. channel(s) to display . number of channels	9. Using a pixel's value from each of two specified channels as indices to the transformation matrix, pick up Laplacian value (LAPL) for that location.	

*Determine: n = number of pixels (columns)/line
groups of n pixels = number of lines in window

Author: ______ Date: ______

Diagram ID: 2.7.3.4 Name: CLRDI6 Description: DISPLAY LAPLACIAN

Input	Process	Output
. color table . intensity table	10. Using value LAPL, determine color and brighness from the color and intensity tables and store in RAMTEK buffer.	. RAMTEK refresh memory buffer
	11. Add 1 to count of frequency of this Laplacian value.	
	12. Repeat steps 7 through 11 for n pixels.	
	13. Repeat steps 8 through 12 for all groups of n pixels.	
	14. Move data from local frequency table to global table.	. frequency table
. type of display . window packet . type of channel	15. Save current display type window packet, and channel type.	. previous type of display . previous window packet . previous type of channel

Author: ____________ Date: ____________

Diagram ID: 2.7.3.5 Name: ____________ Description: DISPLAY OF VARYING COLOR INTENSITY (POLAR DISPLAY)

Input

. window packet (MSAOWW)

Source data tape

- Landsat

. tick table

Process

1. If output window is > 512 by 512 pixels, crop to that size.*
2. Initialize read window and position tape at top of window (according to sample).
3. Zero local frequency table.
4. Place no data symbols in output buffer.
5. Read, unpack, and transform the data for n pixels into two values: intensity and color angle.
6. Normalize intensity values to a range of 0 to 31.
7. Normalize color angle values to a range of 0 to 63.
8. If a tick should be output at current pixel location, store the tick information.
9. If current pixel is not in output window, go to step 14.

Output

. output tick buffer

*Determine: n = number of pixels (columns)/line
groups of n pixels = number of lines in window

Author: ____________ Date: ____________

Diagram ID: 2.7.3.5 Name: ____________ Description: DISPLAY OF VARYING COLOR INTENSITY (POLAR DISPLAY)

Input	Process	Output
. channels to display . radiance ranges for channels	10. If normalized pixel value of intensity (polar channel 1) is not within valid range, ignore and go to step 14.	
	11. If normalized pixel value of the color angle (polar channel 2) is not within valid range, ignore and go to step 14.	
. color table . intensity table	12. Using the color and intensity tables, assign a value to the current pixel based on the brightness (polar channel 1) and color angle (polar channel 2) and store value in RAMTEK refresh memory buffer.	. RAMTEK refresh memory buffer
	13. Add 1 to count of frequency of the discrete color assigned to this pixel.	
	14. Repeat steps 7 through 13 for n pixels.	
	15. Repeat steps 5 through 14 for all groups of n pixels.	
	16. Move data from local frequency table to global table.	. frequency table

Author:

Diagram ID: 2.7.3.5 Name:

Date:

Description: DISPLAY OF VARYING COLOR INTENSITY (POLAR DISPLAY)

Input

- type of display
- window packet
- type of channel

Process

17. Save current display type, window packet, and type of channel for later checking.

Output

- previous type of display
- previous window packet
- previous type of channel

Author: ______

Date: 03/08/79

Diagram ID: 2.7.3.6

Name: CALCOL

Description: SET UP RAMTEK LOOKUP TABLE AND DETERMINE WHETHER SPATIAL CONTENT OF THE DATA HAS BEEN CHANGED

Input

. type of channel
. previous type of channel
. number of channels

. number of channels
. initial color table
. initial intensity table

Process

1. If no display done previously, skip to step 4.

2. If the previous channel type is different from the present channel type, skip to step 4.

3. If display type is significantly different from previous display type, skip to step 4.

4. If number of channels requested is one:

   a. Set up single channel color and intensity tables.

   b. Set up single channel RAMTEK lookup table.

6. If previous display type is RADIANCE and present type is CLASS:

   a. Set up the color table and intensity table for each partition in the label table to mean values in color and intensity tables.

   b. Set up the multichannel RAMTEK lookup table.

Output

. color symbol table
. intensity table

RAMTEK LUT buffer

Author:

Date: 03/08/79

Diagram ID: 2.7.3.6

Name: CALCOL

Description: SET UP RAMTEK LOOKUP TABLE AND DETERMINE WHETHER SPATIAL CONTENT OF THE DATA HAS BEEN CHANGED

Input	Process	Output
	c. Return. 7. Do the following functions and return: a. Set up the color table and the intensity table. b. Set up the multi-channel RAMTEK lookup table.	

Author:

Date:

Diagram ID: 2.7.3.7 Name: SETMEM Description: DETERMINE IS SPATIAL CONTENT OF DATA IS CHANGED

Input	Process	Output
. previous type of display	1. If no display has been done previously, skip to step 6.	
. previous type of channel . channel type	2. If previous channel type is different from present channel type, skip to step 6.	
. display type	3. If present display type changes the spatial content of the data, skip to step 6.	
. previous window packet . window packet	4. If present window definition is different from the previous window definition, skip to step 6.	
	5. Return.	
	6. Set refresh memory flag and return.	. refresh memory flag

C-3

Author: ______________

Date: 01/18/79

Diagram ID: 2.7.4 Name: DISTAB Description: TABULATION OF RADIANCE VALUES, SYMBOLS, NUMBER OF PIXELS

Input

. display type (KLSTYP)

. type of table (KTBLTY)

Card image

. current window number (NWNDOW)

. window from frequency table (KTBLNW)

Process

1. Print message and error flag if no DISPLAY done previously. Determine by looking at display type and presence of frequency table.
2. Get and save type of tabulation. If invalid, print a message and set error flag.
3. If any diagnostics, return.
4. Save current window number and window number from frequency table.
5. Put page and window headings (HDUNIT) in output buffer.
6. Restore current window.
7. Print table headings, if batch.
8. Initialize subtotals, totals, and limits.
9. Print tabular data.
10. Write totals.
11. If error occurred, print message.
12. Blank command word.

Output

. error flag

Error message

. type of tabulation (KTBTYP)

. current window number (NWNDOW)

Tabulation:

- Headings
- Tabular
- Totals

Totals:

. KFRSUB

. KFRTOT

. MINRAD

. MINSYM

. MINCHV

. MAXCHV

Warning message

. command word (KOMD)

Author: 　　　　Date:

Diagram ID: 2.7.4　　Name: DISTB　　Description: TABULATION OF VALUES, SYMBOLS (COLORS), NUMBER OF PIXELS

Input

. display type

. type of table

Card image

. data checkout switch (MCHECK)

. current window number

. window from frequency table

. symbol table or

. color symbol table

. frequency table

Process

1. If no DISPLAY done previously, print message and set error flag.
2. Get and save type of tabulation: SYMBOL, RADIANCE, GRADIENT, LAPLACIAN, CLASS.
3. If type of tabulation is not equal to type of display or not equal to SYMBOL, print a message and set an error flag.
4. If any errors occurred, skip to step 13.
5. If data checkout mode, return.
6. Save current window number and window number from frequency table.
7. Print ID information.
8. Print columnar headings.
9. Restore current window.
10. Initialize subtotals and totals for frequencies.
11. If type of tabulation is SYMBOLS:

Output

Error message

. error flag

. type of tabulation

Tabulation:

- ID information
- Columnar headings

. current window number

Author: ______ Date: ______

Diagram ID: 2.7.4 Name: DISTAB Description: TABULATION OF DISPLAY VALUES, SYMBOLS (COLORS), NUMBER OF PIXELS

Input	Process	Output
	a. Cumulate subtotal frequencies per symbol.	
	b. Cumulate total of all frequencies.	
	c. Print subtotals per symbol.	Tabulation: - Symbols and subtotals
	d. Print total of all frequencies.	- Total
	12. If type of tabulation is not SYMBOLS:	
	a. Cumulate total of all frequencies.	
	b. Print each entry in frequency table with its associated index value and symbol.	Tabulation: - Symbols and frequency
	c. Print total of all frequencies.	- Total
	13. Check error flags and mode switches for printing completion or non-completion messages.	Messages
	14. Blank command name.	. command name (KOMD)

Author: ______ Date: 01/19/79

Diagram ID: 2.7.5 Name: KMDFAC Description: FACTOR RAW MSS CHANNELS

Input

Card image

. output window packet

. input window packet

Process

1. Get and check type of factoring: RADIANCE (default).
2. If type of factoring is invalid, print a message and set error flag.
3. Print message if any other information on card image.
4. Calculate the printer/plotter (PPD) transformation coefficients and then, using those coefficients, calculate the PPD envelope for the current window, then the geographic envelope and the scanner envelope around that PPD envelope.
5. Crop the output window to fit the input window.
6. If first window, print window heading.
7. If any errors occurred, return.
8. On raw or transform channels, do a factor analysis to determine two orthogonal axes within the window (2.7.5.8).

Output

Error message

. error flag

Window packet coordinates:

. adjusted scanner (MSAOWW)

. geographic (GEDOWW)

. UTM (UTMOWW)

. printer/plotter (PPDOWW)

. output window packet

Window heading

.table for factors

Author:

Date: 01/18/79

Diagram ID: 2.7.5.8

Name: KMDFA3

Description: FACTOR RAW MSS CHANNELS (PHASE 3)

Input

- table type (KTBLTY)
- tape unit (LU3FMT)

Process

1. Flag KTABLE as destroyed (KTBLYT = 0).
2. If there is an input tape:
   a. Compute number of pixels, channels, and factors for MSS data.
   b. Compute sums and sums of products for MSS data.
3. If no input tape:
   a. Set number of pixels, channels, factors.
   b. Compute sums and sums of products.
4. Compute correlations, means, SD's.
5. Duplicate correlation matrix.
6. Compute eigenvalues and eigenvectors and sort in order of descending eigenvalues.
7. Compute percent of variance accounted for by each factor.

Output

- table type (KTBLTY)
- number of pixels (KPIXCO)
- number of channels (KCHACO)
- number of factors (KFAPCO)
- table for frequency and factors (KTABLE)

Author:

Date: 01/18/79

Diagram ID: 2.7.5.8   Name: KMDFA3   Description: FACTOR RAW MSS CHANNELS (PHASE 3)

Input	Process	Output
	8. Compute factor structure [the correlation matrix between channels (rows) and factors (columns)]. 9. Compute normalized factor coefficients. 10. Mark KTABLE as containing factor parameters from principal factor analysis.	. table window number (KTBLNW)

Author: ____________

Date: 01/18/79

Diagram ID: 2.7.6 Name: KMDROT Description: ROTATES PREVIOUSLY GENFRATED FACTOR STRUCTURES AND/OR COEFFICIENTS

Input	Process	Output
. table type (KTBLTY) . class type (KLSTYP) Card image	1. If factor table has not been previously loaded, print warning message. 2. If an explicit rotation angle in degrees has been specified: a. Get angle and print a message if it is not between -360° and +360°. b. Pick up optional final angle (default is same as initial). c. Pick up optional increment (default is 1°). d. Perform explicit rotation angle and increment. e. Print rotation angle, final angle, and increment. f. If errors encountered, go to step 6. g. Print structure information and go to step 6.	Warning message . structure or coefficient information (KTABLE) Rotation angle Final angle Increment Structure information

Author: ____________

Diagram ID: 2.7.6

Name: KMDROT

Date: 01/18/79

Description: ROTATES PREVIOUSLY GENERATED FACTOR STRUCTURES AND/OR COEFFICIENTS

Input

Process

3. If not angle in degrees:

    a. Get type of matrix used in evaluating function (default = STRUCTURE).

    b. Get normalized or un-normalized matrix type (default is NORM).

    c. If error occurred, set error flag; go to step 6.

    d. If QUARTIMAX is matrix type

        (1) If STRUCTURE, do rotation of factor structure.

        (2) If not STRUCTURE, do rotation of coefficients.

    e. If VARIMAX is matrix type

        (1) If STRUCTURE, do rotation of factor structure.

        (2) If not STRUCTURE, do rotation of coefficients.

    f. Do 2.e. and 2.f.

Output

Author:

Date: 01/18/79

Diagram ID: 2.7.6 Name: KMDROT Description: ROTATES PREVIOUSLY GENERATED FACTOR STRUCTURES AND/OR COEFFICIENTS

Input	Process	Output
	4. Print factor structure, coefficients, and means.	Structure, coefficients, means
	5. Assign factor coefficients to linear transformation weights and update weighted gains.	. weighted gains
	6. Print messages according to types of errors.	
	7. Clear warning counters.	
	8. Blank user command.	. command word (KOMD)

Author:

Date: 01/22/79

Diagram ID: 2.7.7 Name: KMDPAR Description: CONTROL PARTITIONING OF FACTOR SPACE

Input	Process	Output
Card image	1. Get criterion for partitioning: DENSITY, GRADIENT, LAPLACIAN.	
	2. Print message if any other information on card image.	Warning messages
. low radiance range (LCVLO1) . high radiance range (LCVHI1)	3. Print warning message if no radiance limits were previously specified.	
. data checkout mode switch (MDATAC)	4. If in data checkout mode, return.	
	5. Calculate the printer/plotter (PPD) transformation coefficients and then using these coefficients calculate the PPD envelope for the current window, then the geographic envelope and the scanner envelope around that PPD envelope.	Window packet coordinates: . adjusted scanner (MSAOWW) . geographic (GEDOWW) . UTM (UTMOWW) . printer/plotter (PPDOWW)
. input window packet . output window packet	6. Crop output window to fit input window.	. output window packet
. diagnostic counter (NDTOTL)	7. If diagnostics, blank command and return.	. command word (KOMD)
	8. Clear window number and zero page number.	. window number (NWNDOW) . page number (NPAGE)
	9. Print heading information.	Heading information

Author:

Diagram ID: 2.7.7 Name: KMDPAR

Date: 01/22/79

Description: CONTROL PARTITIONING OF FACTOR SPACE

Input	Process	Output
	10. Blank command word.	. command word (KOMD)
	11. Dependent on criterion from step 1, partition according to: a. DENSITY (2.6.7.11). b. GRADIENT (2.6.7.12). c. LAPLACIAN (2.6.7.13).	. transformation representation matrix
	12. Fill all the gaps in the transformation representation matrix.	
	13. Focus or blur the probability surface (transformation representation matrix).	
	14. Assign labels to probability surface and spread the labels.	
	15. Save the label lookup table.	Signature File - Label lookup table

Author:

Date: 01/23/79

Diagram ID: 2.7.7.11 Name: KMDPA3 Description: PARTITION BY DENSITY

Input	Process	Output
Source data tape - Landsat . number of channels . starting line . starting sample . number of samples . number of pixels	1. Read, unpack, and transform raw MSS data into transformed channels for n pixels. 2. Using a pixel's value from each transform channel (F_1 and F_2) as indices to a matrix, add 1 to that matrix location as an occurrence counter. 3. Repeat steps 1 and 2 for all groups of n pixels until all pixels have been processed.	. transformation representation matrix (counts)

Author: ______

Diagram ID: 2.7.7.12 Name: KMDPA4 Date: 01/11/79

Description: COMPUTE GRADIENT (Find maximum directional derivative)

Input

Source data tape:
- Landsat

. number of channels

. range of lines*

. range of samples*

Process

1. Read, unpack, and transform raw MSS data into transformed channels for n pixels.

2. In a channel, take the radiance value of each of a pixel's neighbors (maximum of 8) and do the following calculation:

$$GRAD = |(A + B + C) - (G + H + I)| + |(A + D + G) - (C + F + I)|$$

where the pixels are arranged as follows:

A	B	C
D	E	F
G	H	I

E is center pixel.
A, B, C, D, F, G, H, and I are nearest neighbors.

3. Add value GRAD to transformation representation matrix using pixel E's value from each transform channel (F_1 and F_2) as the indices to the matrix as follows:

Output

. radiance buffer (IRADBF)

. transformation representation matrix (sums of GRADS and counts)

*Range of lines and range of samples is used to calculate the number of groups of n pixels for processing.

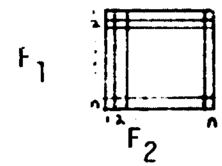
Author:

Date: 01/11/79

Diagram ID: 2.7.7.12 Name: KMDPA4 Description: COMPUTE GRADIENT (Find maximum directional derivative)

Input

Process



4. Add 1 to counter of number of GRAD values added to that specific matrix location (for first channel GRAD calculations)

5. Repeat steps 2 through 4 for n pixels in current channel.

6. Go to next channel and repeat steps 2, 3, and 5.

7. For the next group of n pixels repeat steps 1 through 6 until all pixels have been processed.

8. Calculate mean of GRAD values in each location of transformation matrix.

NOTE: Means and counts are made available as input for subroutine GAPFIL which will fill in gaps and write the completed mean probability surface estimate to disk.

Output

. means of GRAD values in transformation matrix.

Author:

Diagram ID: 2.7.7.13 Name: KMDPA5

Date: 01/15/79

Description: COMPUTE LAPLACIAN (find magnitude of change between pixel and nearest neighbors)

Input

Source data tape:
- Landsat

. number of channels

. range of lines*

. range of samples*

Process

1. Read, unpack, and transform raw MSS data into transformed channels for n pixels.

2. Using only one channel, take the radiance value of a pixel and its neighbors and do the following calculation:

   $$LAPL = |8 * E - (A + B + C + D + F + G + H + I)|$$

   where pixels are arranged as follows:

A	B	C
D	E	F
G	H	I

   E is center pixel. A, B, C, D, F, G, H, and I are nearest neighbors.

Output

. radiance buffer (IRADBF)

*Range of lines and range of samples is used to calculate the number of groups of n pixels for processing.

Author:

Date: 01/15/79

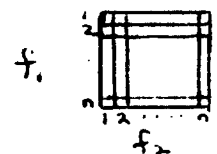
Diagram ID: 2.7.7.13 **Name:** KMDPA5 **Description:** COMPUTE LAPLACIAN

Input

. number of pixels

Process

3. Add value LAPL to transformation representation matrix using pixel E's value from each transformed channel (f_1, f_2) as the indices to the matrix as follows:



4. Add 1 to counter of number of LAPL values added to that specific matrix location (for first channel LAPL calculations).
5. Repeat steps 2 through 4 for n pixels in the current channel.
6. Go to next channel and repeat steps 2, 3, and 5.
7. Repeat steps 1 through 6 for next group of pixels until all pixels have been processed.
8. Calculate mean of LAPL values in each location of transformation matrix.

Output

. transformation representation matrix (sums of LAPL's and counts)

Author: ____________

Diagram ID: 2.7.8 Name: KMDHIS Description: HISTOGRAM OF FREQUENCY Date: 02/20/79

Input	Process	Output
. type of table	1. Check if frequency table is loaded. If not, set error flag.	. error flag
	2. Get and save type of histogram. If invalid, set error flag.	. type of histogram
	3. If errors, print message(s) and return.	Error messages
	4. Save current window number and replace with previous window number.	
. output device (Indicated by name of the calling program: PRTCLASS CRTCLASS PLTCLASS)	5. Build histogram heading and write to printer, CRT or plotter, depending on output device.	Histogram - Heading
	6. Restore current window number.	
. frequency table	7. Calculate scaling of histogram for both the symbol axis and the frequency axis. The scaling depends on size of output device, difference between smallest and largest frequency values in the table, and two constants:	.scaling factors

Author: ____________

Diagram ID: 2.7.8 Name: KMDHIS Description: HISTOGRAM OF FREQUENCY

Date: 02/20/79

Input	Process	Output
	a. The height of the histogram will always be spread over the same part of the CRT screen, printer page, or plotter sheet, leaving standard places for headings and bottom scale.	
	b. There will always be 60 bins, or columns, in the histogram. The number of symbols in the display or map may be less than that, meaning some bins will be empty.	
	8. Calculate an appropriate interval for the ticks and labels on the frequency axis.	. frequency axis tick interval
. frequency table . symbol table	9. Calculate the total number of pixels found for each symbol in the symbol table and store totals in bin array.	. bin array
	10. Calculate the number of bins used out of the possible 60 and initialize the bin array pointer to one.	. number of bins used . bin array pointer
. output device	11. If output device is printer, calculate and plot histogram as follows:	

Author: ______________

Diagram ID: 2.7.8 Name: KMDHIS Description: HISTOGRAM OF FREQUENCY

Date: 02/20/79

Input	Process	Output
	a. Calculate the largest frequency to be labeled on the frequency axis, and store it as the starting frequency level for processing the histogram, which will be generated from the top down.	. frequency level
	b. If the value in the current bin of the bin array is greater than or equal to the current frequency level, fill the corresponding space in output line buffer with appropriate symbol.	. output buffer
	c. Increment the bin array pointer and repeat from step b for each bin used.	
	d. Pad the left of the output buffer with blanks.	
	e. If current frequency level is evenly divisible by the frequency axis tick interval, insert frequency number and tick mark on left axis.	
	f. Print the output buffer.	Histogram - Bins - Left-column scale

Author:

Diagram ID: 2.7.8 Name: KMDHIS Date: 02/20/79

Description: HISTOGRAM OF FREQUENCY

Input	Process	Output
. frequency axis tick interval . number of bins used . symbol table or . color symbol table	g. Reinitialize bin array pointer and repeat from stcp b for each line in the histogram, decrementing the current frequency level by a calculated scaling factor. h. Write bottom scale. 12. If output device is plotter or CRT, calculate and plot histogram as follows: a. Position pen or cursor; plot frequency axis and symbol axis of histogram. b. Plot marginal ticks for frequency axis and label. c. Plot 60 marginal ticks for symbol axis and label columns used. d. Position pen or cursor at histogram origin. e. Plot vertically to frequency level indicated by current location in bin array, across the width of the histogram column, and back down to symbol axis.	Histogram - Bottom scale - Frequency axis scale - Symbol axis scale - Bins

Author:

Date: 02/20/79

Diagram ID: 2.7.8

Name: KMDHIS

Description: HISTOGRAM OF FREQUENCY

Input	Process	Output
	f. If output device is CRT, fill area just outlined (bin) with current color. g. Increment bin array pointer. h. Repeat from step e for every bin used. 13. Blank command word.	. command word

Author: ______ Date: 02/10/79

Diagram ID: 2.7.9 Name: PROFILE Description: DISPLAY A PROFILE OF RADIANCE VALUES SPECIFIED BY LINE SEGMENT(S)

Input	Process	Output
Card image Source data tape: - Landsat	1. Get all starting and ending coordinates of points in a display defining the line segments from which a profile is to be constructed. 2. Get and check range of profile plot and print a message if error (default = 0 - 127). 3. Determine maximum and minimum line and samples to retrieve for processing. 4. Rewind tape. 5. To prepare for output, compute printer/plotter spacing and compute the envelopes in needed coordinates. 6. Initialize page number, window number, and header buffer. 7. Initialize read window and position tape at top of window according to line and sample.	. coordinates of profile segments . profile plot range . error flag Error message . output window packet . page number . window number . header buffer

Author:

Date: 02/10/79

Diagram ID: 2.7.9

Name: PROFILE

Description: DISPLAY A PROFILE OF RADIANCE VALUES SPECIFIED BY LINE SEGMENT(S)

Input

- type of channel
- channel to display
- radiance ranges for specified channels
- number of channels

Process

8. Zero local frequency table.
9. Read, unpack, and, if applicable, transform data for n pixels in all channels.
10. If current pixel is not part of any line segment defining the profile, go to step 15.
11. If the radiance value of current pixel is within the specified range for the current channel, save value in output buffer for later display.
12. If radiance value is out of range high, store high value in pixel location.
13. If radiance value is out of range low, store low value in pixel location.
14. Add 1 to frequency count for this radiance value.
15. Repeat steps 11 through 13 for all specified channels.
16. Repeat steps 9 through 14 for n pixels.

Output

- output buffer

Author:

Diagram ID: 2.7.9 Name: PROFILE Date: 2/10/79

Description: DISPLAY A PROFILE OF RADIANCE VALUES SPECIFIED BY LINE SEGMENT(S)

Input	Process	Output
	17. Repeat steps 8 through 15 for all groups of n pixels.	
	18. Move data from local frequency tables into global table. →	. frequency table
	19. Output header buffer, tick intervals, and output buffer to color graphics (2.15). →	Display

Author: ______

Date: 03/08/79

Diagram ID: 2.7.10

Name: CLREXI

Description: EXIT FROM CLRTAB

Input	Process	Output
. data/checkout flag Source data tape: - Landsat . fatal error flag	1. If in data checkout mode, exit. 2. Close Landsat tape file. 3. If fatal errors occurred, and user does not want displays printed, delete any print files. 4. If fatal errors did not occur, or fatal errors occurred and user wants displays printed anyway, close print files. 5. Exit.	

Author: RSIS

Date: ______

Diagram ID: 2.8

Name: CLASSIFY

Description: CLASSIFY DATA ACCORDING TO SPECTRAL LIMITS

Input

Registration Parameter File

Signature File:
- Label lookup table

Source data tape:
- Landsat

Card images

Process

1. Initialize by opening files, loading registration parameters, identifying Landsat scene, initializing spacing, and queing default commands (2.8.1).
2. Read and store command.
3. If command defines or selects specifications, store specifications in applicable variables (2.8.2).
4. If command is DETECT, the program detects all occurrences of the previously specified spectral limits within the current window and stores this information in a Detection File (2.8.3).
5. If command is EXIT, terminate program (2.8.4).
6. Repeat steps 3 through [illegible] until program terminates.

Output

. command name

. specification variables

Detection File:
- Density
- Radiance
- Class

Author: ______

Date: 01/25/79

Diagram ID: 2.8.1 Name: CLAXQT Description: INITIATE CLASSIFY

Input	Process	Output
Signature File: - Label lookup table Source data tape: - Landsat Registration Parameter File	1. Print program ID. 2. Initialize Log File. 3. Open Signature File. 4. Open Landsat File and save short scene ID. 5. Load registration parameters and complete scene ID from Adjusted Control Point Network File. 6. Print short ID for Landsat scene. 7. Print complete ID for Landsat scene. 8. Store generation date of Detection File. 9. Initialize spacing to 1 line and 1 sample. 10. Queue default commands.	Program name/date/time Short Landsat ID Complete Landsat scene ID Log File: - Initialized log records . other scene information . Landsat number (NERTS) . CCT number (NCCT) . registration parameters . generation date (JENMDT) . line length adjusted MSS coordinates (MSAOWW)

Author:

Date: 02/16/79

Diagram ID: 2.8.2

Name:

Description: COMMANDS THAT SPECIFY

Input	Process	Output
Card image . command name	1. If command is CHANNEL, get channel number and type of channel to display or process (2.7.2.2).	. channel number . type of channel
	2. If command is NAME, get name of transform or spectral limits (2.8.2.3).	. name
	3. If command is LINEAR, get transformation coefficients to be used in subsequent processing of linear channels (2.7.2.4).	. linear gain . linear weight . weighted gain
	4. If command is ORIGIN, get coordinate system and coordinates of the origin of the scene (2.7.2.5).	. window packet coordinates . system of coordinates
	5. If command is RADIANCE, get radiance limits or range for channel specified by CHANNEL command (2.7.2.6).	Radiance Range: . minimum radiance . maximum radiance
	6. If command is POLAR, specify the gain(s) and bias(es) to be used in processing polar channel(s): composite intensity channel and color angle channel (2.7.3.5).	Polar channel 1: . gain . bias Polar channel 2: . gain . bias

Author:

Date: 02/16/79

Diagram ID: 2.8.2

Name:

Description: COMMANDS THAT SPECIFY

Input	Process	Output
	7. If command is WINDOW, get the coordinate system and coordinates of that system for the vertices of a window (2.7.2.9).	Window Packets: . coordinate vertex system . vertices
	8. If command is ZONE, get the universal transverse mercator (UTM) zone number to be used in subsequent processing of UTM coordinates (2.7.2.10).	. UTM central meridian . UTM zone number

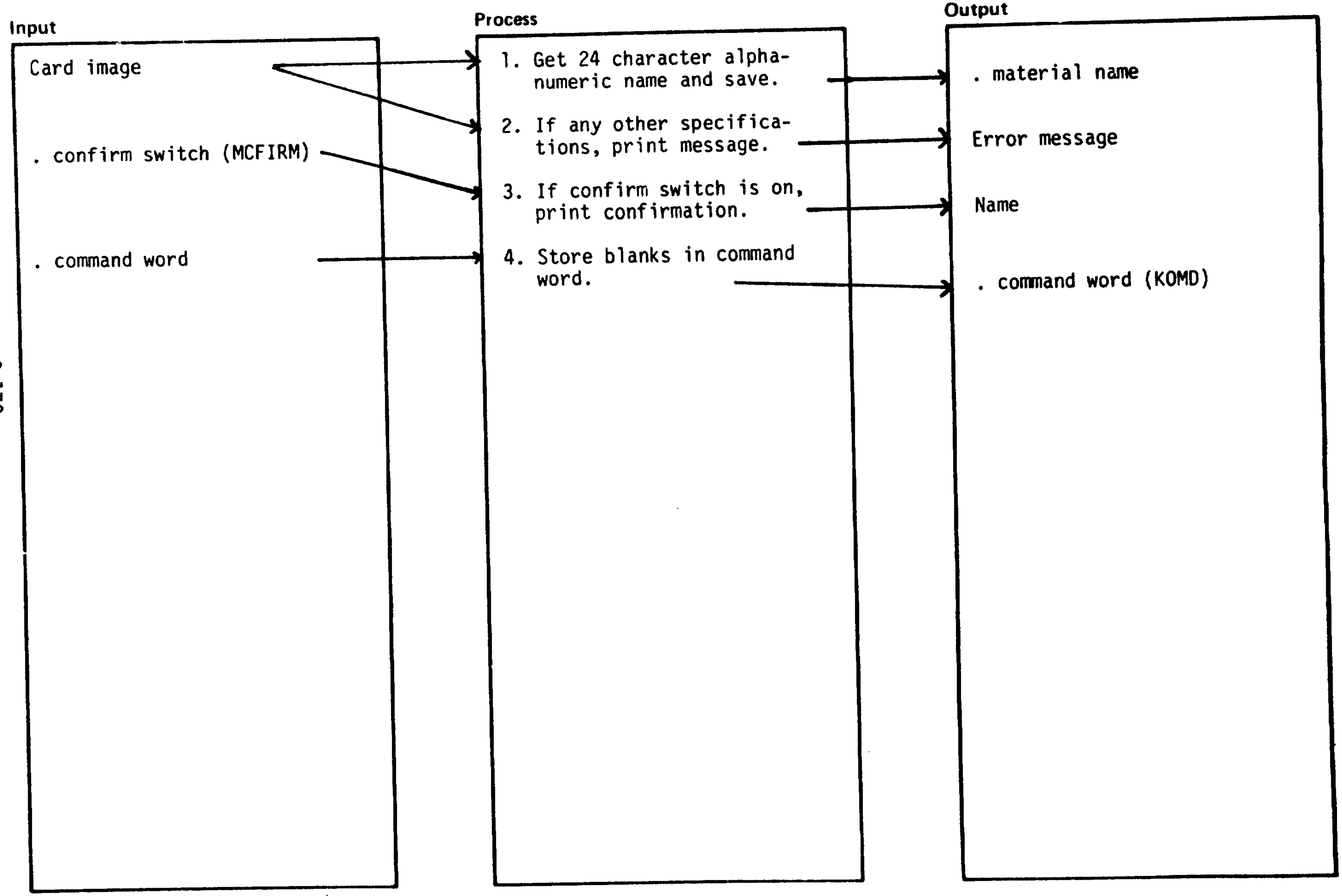
Author:

Date:

Diagram ID: 2.8.2.3

Name:

Description: PROCESS NAME COMMAND



Author: ______ Date: ______

Diagram ID: 2.8.3 Name: CLADET Description: CONTROL GENERATION OF DETECTION FILES

Input

. window number (NWNDOW)

Card image

Signature File
- Label lookup table

. data checkout switch (MDATAC)

Process

1. If window is not specified, print message and set error flag.

2. Get and save type of Detection File: CLASS, DENSITY, RADIANCE. If invalid type, print error message and set error flag.

3. Print a message if any other specification field on card image.

4. If type of file is CLASS or RADIANCE:

   a. If a Signature File is not present, write a message and set error flag.

   b. If Signature File is present, read it and store label lookup table.

5. If data checkout switch is on, return.

6. If any errors occurred, return.

Output

Error message

. error flag

. type of Detection File

. label lookup table

Author: ____________ Date: ____________

Diagram ID: 2.8.3 Name: CLADET Description: CONTROL GENERATION OF DETECTION FILES

Input	Process	Output
. input window packet (MSAIWW) . output window packet (MSADWW) . envelope minimum pointer (WMIN) . envelope maximum pointer (WMAX)	7. Clear window and reset page number. 8. Plot spectral information on printer. 9. Compute printer/plotter spacing and compute envelope in needed coordinates. 10. Eliminate any part of MSS output window outside of MSS input window. 11. Initialize detection output window. 12. If any errors occurred previously, exit with a call to close files. 13. Open Detection File if it does not already exist (2.8.3.1). 14. If type of Detection File is DENSITY, generate density file for water (2.8.3.2). 15. If type of Detection File is RADIANCE, generate radiance file (2.8.3.3). 16. If type of Detection File is CLASS, generate a class file (2.8.3.4).	. window number (NWNDOW) . page number (NPAGE) Spectral information Envelopes: . PPDOWW . UTMOWW . GEDOWW . output window packet (MSAOWW) . detection output window (MSADWW) Detection File: - Density - Class - Radiance

Author: ______

Date: 1/25/79

Diagram ID: 2.8.3.1 Name: OPN02N Description: OPEN/CHECK OUTPUT DETECTION FILE

Input

Detection File

. Landsat number
. CCT number
. detection output window (MSADWW)
. other scene information

Process

1. Assign new Detection File to this run, if old one does not exist. Skip to step 6.
2. Assign old Detection File, if it exists.
3. Read old Detection File header into buffer.
4. Compare old Detection File header information with the scene information obtained from the Adjusted Control Network File and requested window information. If they are equal, return.
5. If old Detection File is not the same window as requested, print a message.
6. Write interim blank header to either replace existing Detection File or put in new file.

Output

Error message

Detection File

Author:

Date: 01/25/79

Diagram ID: 2.8.3.2 Name: CLADE3 Description: GENERATE DENSITY FILE

Input

. density output window header packet (MSADWW)

. buffer number for limit channel (LIMCH)

Source data tape:

- Landsat

. limit values (LIMVAL)

Process

1. Initialize window, channel switches, unpacking parameters, density fill packet, and buffer assignments, buffers.
2. Read Landsat MSS data (GETRAD).
3. Classify pixel "hit" within limits.
4. Store classification data for hits.
5. Classify pixel "near hit".
6. Store classification data for near hits.
7. Rotate detection record buffers.
8. Write the classification density record on disk in a packed form for subsequent use by PRTCLASS and PLTCLASS programs.

Output

Detection File:

- Density

Author:

Diagram ID: 2.8.3.4 Name: CLADE5 Description: GENERATE CLASS FILE

Date: 02/02/79

Input

Source data tape:
- Landsat

. type of channel

. channels to display

. label lookup table

. number of pixels

Process

1. Initialize unpacking parameters.
2. Read, unpack, and transform (if applicable) the data (2.6.3.1.8).
3. Using a pixel's value from each transform channel (F_1 and F_2) as the indices to the label lookup table, pick up label value and store in output buffer.
4. Repeat step 3 for n pixels.
5. Write the class output buffer onto the class Detection File for subsequent use by CRTCLASS, PRTCLASS, PLTCLASS, or FLMCLASS programs.
6. Clear output buffer.
7. Repeat steps 3 through 6 for all groups of n pixels.

Output

Detection File:
- Class

Author: ____________ Date: ____________

Diagram ID: 2.8.3.8 Name: CLADE4 Description: GENERATE RADIANCE FILE

Input

Source data tape:
- Landsat

. type of channel

. channel to output

. ranges for channels

. number of pixels

Process

1. Initialize unpacking parameters and initialize output buffer.
2. Read, unpack, and transform (if applicable) the data (2.6.3.1.8).
3. If a pixel's radiance value from the output channel is not within range specified, skip the pixel.
4. If a range is specified for another channel, check the pixels value and if not within that channel range, skip the pixel.
5. Store the radiance value from output channel in output buffer.
6. Repeat steps 3 through 5 for n pixels.
7. Write the radiance output buffer onto the Radiance File.
8. Clear output buffer.
9. Repeat steps 2 through 8 for all groups of n pixels.

Output

Detection File:
- Radiance

Author:

Date: 12/14/78

Diagram ID: 2.8.4 Name: CLAEXI Description: TERMINATION ROUTINE FOR CLASSIFY

Input

- data/checkout mode switch (MDATAC)
- window number (NWNDOW)

Process

1. Print program termination message.
2. If data/checkout mode, skip to step 6.
3. Close and verify EOF file on Landsat tape.
4. If window number not equal to zero, write detection header record and close output detection file (2.8.4.2).
5. Close alternate print files.
6. Terminate program.

Output

Termination message

Detection File

Author:

Date: 01/25/79

Diagram ID: 2.8.4.2

Name: CLOS2N

Description: WRITE DETECTION HEADERS AND CLOSE DETECTION FILE

Input

- number of CCT's
- fatal error flag
- other scene information
- Landsat number (NERTS)
- CCT number (NCCT)

Process

1. If no CCT's or too many CCT's, return.
2. If any fatal errors occurred, do not write headers.
3. If no errors occurred in CLASSIFY, move scene information, Landsat number, and CCT number into header output buffer.
4. Write header onto Detection File.
5. Catalog Detection File.

Output

Detection File

Author:

Date: 03/05/79

Diagram ID: 2.9 Name: PRTDET Description: PRINT DETECTION FILE

Input

Detection File

Process

1. Initialize by opening files, identifying Landsat scene, and queuing default commands (2.9.1).
2. Read and save commands.
3. If command defines or selects specifications, store specifications in applicable variables (2.9.2).
4. If command is DISPLAY, display Detection File using symbols (2.9.3).
5. If command is LIST, print values in each scan line (2.9.4).
6. If command is EXIT, close files and terminate program (2.9.5).
7. If invalid command, write a message.
8. Repeat steps 2 through 6.

Output

. command name (KOMD)

. specification variables

Display:
- Class
- Density
- Radiance

List:
- Class
- Density
- Radiance

Error message

Author:

Date: 03/05/79

Diagram ID: 2.9.1

Name:

Description: OPEN FILE AND IDENTIFY

Input	Process	Output
Detection File: - Class - Density - Radiance	1. Open Detection File, read header, and store information. 2. Print header information to identify file type.	. type of Detection File Identification of Detection File

Author: ____

Date: 03/05/79

Diagram ID: 2.9.2

Name: ____

Description: PROCESS COMMANDS THAT SPECIFY

Input	Process	Output
. command word (KOMD) Card image	1. If command is CHANNEL, get channel number and type of channel (2.7.2.2).	. channel to display
	2. If command is CLASS, get class limits or range of values to display (2.10.2.6).	. class range
	3. If command is DENSITY, get density limits or range of values to display (2.10.2.1).	. density range
	4. If command is RADIANCE, get radiance limits or range of values to display (2.7.2.6).	. radiance range
	5. If command is SPACING, get the line and sample increments to be used for processing subsequent windows (2.7.2.7).	. line increment . sample increment
	6. If command is SYMBOL, get symbol and range of values assigned that symbol (2.6.2.8).	. symbol table
	7. If command is WINDOW, get the coordinate system and coordinates of the vertices of a window (2.7.2.9).	. coordinate vertex system . vertices

Author: ______

Diagram ID: 2.9.3

Name: ______

Date: 03/05/79

Description: DISPLAY SCAN LINES

Input	Process	Output
Card image . type of Detection File	1. Get and save type of Detection File to display.	. type of file to display
	2. If type of file to display is not equal to type of Detection File, print error message and set error flag.	Error message . error flag
	3. If window has not been specified, print error message and set error flag.	
	4. Determine if overprinting is required for output.	
	5. Compute printer/plotter spacing and compute envelope in needed coordinates.	
	6. Crop the output window to fit the input window.	
	7. If errors occurred, skip to step 16.	
	8. Clear window number and reset page number.	. window number (NWNDOW) . page number (NPAGE)
	9. Generate tick intervals.	. tick table
	10. Compute the size of the print window.	
	11. Print window heading.	Window heading
. data checkout switch	12. If data checkout switch is on, skip to step 16.	

Author: ______ Date: 03/05/79

Diagram ID: 2.9.3 Name: ______ Description: DISPLAY SCAN LINES

Input	Process	Output
	13. If type of Detection File to display is CLASS, display class file (2.9.3.14).	Display: - Class
	14. If type of Detection File to display is DENSITY, display density file (2.9.3.15).	- Density
	15. If type of Detection File to display is RADIANCE, display radiance file (2.9.3.16).	- Radiance
	16. Blank command name.	. command name (KOMD)

Author:

Date: 03/09/79

Diagram ID: 2.9.3.14 Name: Description: DISPLAY CLASS FILE

Input

. window packet (MSAOWW)

Detection File:
- Class

. class range

Process

1. Compute size of print window:

   n = number of samples (columns)/line

   X = number of lines in window

   number of pixels = n*X

2. Initialize read window and position file at top of window.
3. Zero local frequency table and initialize output buffer to no data symbols.
4. Generate and print sample scale and border.
5. Initialize output buffer to no data symbols.
6. Read detection line and unpack data for n pixels.
7. If current pixel is not included in the output window, go to step 12.
8. If class value of the pixel does not fall within specified valid range, go to step 12.

Output

Display:
- Sample scale and border

Author: ____

Diagram ID: 2.9.3.14

Name: ____

Date: 03/05/79

Description: DISPLAY CLASS FILE

Input

. symbol table

Process

9. Apply the class value of the pixel to the symbol table to determine the symbol for that pixel.
10. Store the symbol in the output buffer.
11. Add 1 to frequency count of this class.
12. Repeat steps 7 through 10 for n pixels.
13. Print output buffer.
14. Repeat steps 5 through 11 until full print window is output (x lines).
15. Print border and sample scale.
16. Move frequency data into global table.

Output

. output buffer

Display:

- Class (output buffer)
- Border and sample scale

. frequency table

Author: ______________ Date: 03/05/79

Diagram ID: 2.9.3.15 Name: ______________ Description: DISPLAY RADIANCE FILES

Input

. window packet

Detection File:
- Class

. radiance range

Process

1. Compute size of print window:

   n = number of pixels (columns)/line

   X = number of lines in window

   number of pixels = n*X

2. Initialize read window and position file at top of window.
3. Zero local frequency table.
4. Generate and print sample scale and border.
5. Initialize output buffer to no data symbols.
6. Read detection line and unpack data for n pixels.
7. If current pixel is not included in the output window, ignore and go to step 12.
8. If the radiance value of the pixel does not fall within specified valid range, ignore and go to step 12.

Output

Display:
- Sample scale and border

. output buffer

Author: ____________ Date: 03/05/79

Diagram ID: 2.9.3.15 Name: ____________ Description: DISPLAY RADIANCE FILES

Input

. symbol table → 9

Process

9. Apply the radiance value of the pixel to the symbol table to determine the symbol for that pixel.
10. Store the symbol in the output buffer. → output buffer
11. Add 1 to frequency count of this radiance value.
12. Repeat steps 7 through 11 for n pixels.
13. Print output buffer. → Radiance (output buffer)
14. Repeat steps 5 through 11 until full print window is output (x lines).
15. Print border and sample scale. → Border and sample scale
16. Move frequency data into global table. → frequency table

Output

. output buffer

Display:

- Radiance (output buffer)
- Border and sample scale

. frequency table

Author: ______

Diagram ID: 2.9.3.1.6 Name: ______ Description: DISPLAY DENSITY FILE

Date: 03/05/79

Input

. window packet (MSAOWW)

Detection File:

- Density

. density range

Process

1. Compute size of print window:

   n = number of pixels (columns)/line

   X = number of lines in window

   number of pixels = n*X

2. Initialize read window and position file at top of window.
3. Zero local frequency table.
4. Generate and print sample scale and border.
5. Initialize output buffer to no data symbols.
6. Read a detection line and unpack data for n pixels.
7. If current pixel is not included in the output window, ignore and go to step 12.
8. If density value of the pixel does not fall within specified valid range, store a no data symbol in output buffer and go to step 12.

Output

Display:

- Sample scale and border

. output buffer

Author: ____________

Diagram ID: 2.9.3.1.6 Name: ____________ Date: 03/05/79

Description: DISPLAY DENSITY FILE

Input	Process	Output
. symbol table	9. Apply the density value of the pixel to the symbol table to determine the symbol for that pixel.	
	10. Store the symbol in the output buffer.	. output buffer
	11. Add 1 to frequency count of this density value.	
	12. Repeat steps 7 through 11 for n pixels.	
	13. Print output buffer.	Display: - Density (output buffer)
	14. Repeat steps 5 through 13 until full print window is output (x lines).	
	15. Print border and sample scale.	- Border and sample scale
	16. Move frequency data into global table.	. frequency table

Author:

Date: 03/05/79

Diagram ID: 2.9.4

Name:

Description: LIST DETECTION FILES

Input

. window number

Card image

Detection File:
- Class
- Density
- Radiance

Process

1. If window has not been specified, print error message and set error flag.
2. Get type of list and save.
3. If invalid type of list, print error message and set error flag.
4. Compute printer/plotter spacing and compute envelope in needed coordinates.
5. Crop the output window to fit the input window.
6. If error occurred, skip to step 13.
7. Clear window number and reset page number.
8. Compute size of print window in lines and columns.
9. Print window heading.
10. If type of list is CLASS, list a class file (2.9.4.1).
11. If type of list is DENSITY, list a density file (2.9.4.2).
12. If type of list is RADIANCE, list radiance file (2.9.4.3).

Output

Error message

. error flag

. type of list

. window number (NWNDOW)

. page number (NPAGE)

Window heading

List:
- Class
- Density
- Radiance

Author: ____________

Date: 03/05/79

Diagram ID: 2.9.4

Name: ____________

Description: LIST DETECTION FILES

Input	Process	Output
	13. Blank command name. →	. command name (KOMD)

Author:

Date: 03/05/79

Diagram ID: 2.9.4.1 Name:

Description: LIST CLASS FILE

Input

Detection File:
- Class

. class range

Process

1. Compute size of print window:

   n = number of pixels/line

   X = number of lines in window

2. Initialize read window and position file at top of window.
3. Generate and print top sample scale and border.
4. Initialize output buffer to no data symbols.
5. Read detection line and unpack data for n pixels.
6. If current pixel is not included in output window, skip to step 9.
7. If pixel does not fall within valid range, skip to step 9.
8. Encode pixel value to either 2 or 3 characters in output buffer.
9. Repeat steps 6 through 8 for n pixels.

Output

List:
- Top sample scale and border

. output buff.

List:
- Class data

Author:

Date: 03/05/79

Diagram ID: 2.9.4.1

Name:

Description: LIST CLASS FILE

Input	Process	Output
	10. Print output buffer with left and right margins marked.	. Output buffer
	11. Repeat steps 4 through 10 until the complete window is output (x lines).	
	12. Print bottom border and sample scale.	List: - Bottom border and sample scale

Author: ______

Diagram ID: 2.9.4.2

Name: ______

Date: 03/05/79

Description: LIST DENSITY FILE

Input

Detection File
- Density

. density range

Process

1. Compute size of print window:

   n = number of pixels/line

   X = number of lines in window

2. Initialize read window and position tape at top of window.
3. Generate and print top sample scale border.
4. Initialize output buffer to no data symbols.
5. Read detection line and unpack data for n pixels.
6. If current pixel is not included in output window, skip to step 9.
7. If pixel does not fall within valid range, skip to step 9.
8. Encode pixel into 2 or 3 columns depending on its density value.
9. Repeat steps 6 through 8 for n pixels.

Output

List:
- Top sample scale and border
- Density data

Author: ______________ Date: 03/05/79

Diagram ID: 2.9.4.2 Name: ______________ Description: LIST DENSITY FILE

Input

Process

10. Print output buffer with left and right margins marked.
11. Repeat steps 4 through 10 until the complete window is output (x lines).
12. Print bottom border and sample scale.

Output

. Output buffer

List:

- Bottom border and sample scale

Author:

Date: 03/05/79

Diagram ID: 2.9.4.3

Name:

Description: LIST RADIANCE FILE

Input

Detection File

- Radiance

Process

1. Compute size of print window:

   n = number of pixels/line

   X = number of lines per window

2. Initialize read window and position tape at top of window.
3. Generate and print top sample scale and border.
4. Initialize output buffer to no data symbols.
5. Read and unpack detection line for n pixels.
6. If current pixel is not included in output window, skip to step 9.
7. If pixel does not fall within valid range, skip to step 9.
8. Encode pixel into 2 or 3 columns, depending on its radiance value.
9. Repeat steps 6 through 8 for n pixels.

Output

List:

- Top sample scale and border
- Radiance data

Author:

Date: 03/05/79

Diagram ID: 2.9.4.3

Name:

Description: LIST RADIANCE FILE

Input	Process	Output
	10. Print output buffer with left and right margins marked.	. Output buffer
	11. Repeat steps 4 through 10 until the complete window is output (x lines).	
	12. Print bottom border and sample scale.	List: - Bottom border and sample scale

Author:

Date: 02/15/79

Diagram ID: 2.10 Name: PRTCLASS Description: PROGRAM PRODUCE LINE PRINTER MAPS

Input	Process	Output
Detection Files Registration Parameter File	1. Initialize by opening files, loading system date and time, printing program start message, and setting mode switches (2.10.1).	
Card images: - User directives	2. Read commands and specifications.	
	3. If command defines or selects specifications, store them in applicable variables (2.10.2).	. specification variables
	4. If command is AREA, calculate and print area within user-specified closed figure (2.13.4).	Area
	5. If command is LENGTH, calculate and print length of user-specified line segment or series of connected line segments (2.13.5).	Length
	6. If command is MAP, generate registered map(s) (2.10.3).	Maps
	7. If command is TABULATE, print tabulation of frequency (2.6.4).	Tabulations
	8. If command is HISTOGRAM, print histogram of frequency (2.7.5).	Histogram

Author:

Date: 02/15/79

Diagram ID: 2.10

Name: PRTCLASS

Description: PROGRAM PRODUCE LINE PRINTER MAPS

Input	Process	Output
	9. If command is EXIT, exit from PRTCLASS (2.10.4). 10. Print message if illegal command. 11. Repeat from step 2.	Error messages

Author: ____________ **Date:** 02/07/79

Diagram ID: 2.10.1 **Name:** PRCXQT **Description:** INITIALIZE

Input	Process	Output
Detection Files	1. Get system date, time, and mode switches. Print program start message and place message in Log File.	Program start message . mode switches . system date . system time
Registration Parameter File	2. Open Detection Files and identify file types.	Log File . file types . input window packets
	3. Print Landsat scene and load registration parameters.	Scene identification . registration parameters
	4. Initialize default commands.	

Author: ______ Date: 02/15/79

Diagram ID: 2.10.2 Name: ______ Description: PROCESS COMMANDS THAT SPECIFY

Input	Process	Output
. command name Card image	1. If command is CLASS, store range of classes to be used in processing windows (2.10.2.6).	. class range
	2. If command is DENSITY, store density range to be used in processing windows (2.10.2.1).	. density range
	3. If command is HEADING, store text to be used in window headings (2.10.2.2).	. headings
	4. If command is MERIDIAN, store non-standard UTM central meridian (2.10.2.3).	. UTM central meridian
	5. If command is ORIGIN, store window origin (2.7.2.5).	. origin
	6. If command is PRINTER, store printer characteristics (2.10.2.5).	. printer characteristics
	7. If command is RADIANCE, store radiance range to be used in processing windows (2.7.2.6).	. radiance
	8. If command is SCALE, store map scale desired (2.10.2.4).	. scale

Author: ____ Date: 02/15/79

Diagram ID: 2.10.2 Name: ____ Description: PROCESS COMMANDS THAT SPECIFY

Input	Process	Output
	9. If command is SYMBOL, store symbol in appropriate bins in symbol table according to value or range specified (2.6.2.3).	. symbol table
	10. If command is TICK, store tick interval (2.7.2.8).	. tick interval
	11. If command is WINDOW, store window vertices (2.7.2.9).	. window vertices
	12. If command is ZONE, store UTM zone and standard UTM central meridian (2.7.2.10).	. zone . UTM central meridian
	13. If illegal command, print message.	Error message

Author:

Date: 02/15/79

Diagram ID: 2.10.2.1 Name: KMDDEN Description: STORE DENSITY RANGE

Input	Process	Output
Card image	1. Check density range. If invalid, set error flag.	. density range
	2. Calculate count per pixel. If invalid, set error flag.	
	3. If error, print message(s) and return.	Error messages
. confirm switch	4. If confirm switch is on, print density range.	Density range
	5. Blank command word.	. command word

Author:

Date: 02/15/79

Diagram ID: 2.10.2.2 Name: KMDHEA Description: STORE WINDOW HEADINGS

Input	Process	Output
Card image	1. Check that there are not more than two heading lines specified. If error, print message and return.	Error messages . headings
. confirm switch	2. If confirm switch is on, print headings.	Headings
	3. Blank command word.	. command word

Author:

Diagram ID: 2.10.2.3   Name: KMDMER   Date: 02/15/79

Description: STORE UTM CENTRAL MERIDIAN

Input	Process	Output
Card image	1. Check meridian. If invalid, print message and return.	Error messages . UTM central meridian
. confirm switch	2. If confirm switch is on, print meridian.	Meridian
	3. If meridian has been changed, delete origin.	. origin
	4. Blank command word.	. command word

Author: ____________

Diagram ID: 2.10.2.4 Name: KMDSCA Description: STORE MAP SCALE

Date: 02/15/79

Input	Process	Output
Card image	1. Check scale. If out of range, print message and return.	. scale Error message
. confirm switch	2. If confirm switch is on, print scale.	Scale
. window number	3. If scale has changed, flag window as uncalibrated (make window number negative).	. window number
	4. Blank command word.	. command word

Author:

Date: 02/21/79

Diagram ID: 2.10.2.5 Name: KMDPRI Description: STORE PRINTER CHARACTERISTICS

Input	Process	Output
Card image . confirm switch	1. If printer function specified is lines and columns per inch: a. Check lines per inch. If error, set error flag and print message. b. Check columns per inch. If error, set error flag and print message. c. If no errors, store lines and columns per inch. d. If confirm switch is on, print lines and columns per inch. 2. If printer function specified is lines and columns per page: a. Check lines per page. If error, set error flag and print message. b. Check columns per page. If error, set error flag and print message. c. If no errors, store lines and columns per page.	. error flag Error messages . lines per inch . columns per inch Lines and columns per inch . lines per page . columns per page

Author: ______

Date: 02/21/79

Diagram ID: 2.10.2.5 Name: KMDPRI Description: STORE PRINTER CHARACTERISTICS

Input	Process	Output
	d. If confirm switch is on, print lines and columns per page.	Lines and columns per page
	3. If printer function specified is device type:	
	a. Check device type, If error, print message and return.	
	b. Store device type.	. device type
	c. If confirm switch is on, print device type.	Device type
	4. If printer function specified is number of alternate print files:	
	a. Check number of alternate print files. If invalid, print message and return.	
	b. Store number of print files.	. number of alternate print files
	c. If confirm switch is on, print number of alternate print files.	Number of alternate print files
	5. If printer function specified is printer control:	

Author:

Date: 02/21/79

Diagram ID: 2.10.2.5 Name: KMDPRI Description: STORE PRINTER CHARACTERISTICS

Input	Process	Output
	a. If type of control is not AUTO or MANUAL, print message and return.	
	b. Store type of control. →	. type of printer control
	c. If confirm switch is on, print type of printer control. →	Type of printer control
	6. Blank command word. →	. command word

Author:

Diagram ID: 2.10.2.6 — Name: KMDCLA — Date: 02/28/79 — Description: STORE CLASS RANGE

Input	Process	Output
Card image	1. Get and check range. If invalid, print message and return.	Error message . class range
. confirm switch	2. If confirm switch is on, print class range.	Class range
	3. Blank command word.	. command word

Author:

Diagram ID: 2.10.3 Name: PRCMAP

Date: 02/09/79

Description: GENERATE REGISTERED MAPS

Input	Process	Output
. number of subwindows	1. Get/check the number of subwindows. If not valid, set error flag.	. error flag
. symbol table	2. Complete the symbol table by calculating the maximum number of overprint characters needed to print any symbol.	. symbol table
. scale	3. Calculate mapping equations for scale.	. transformation coefficients from corrected scanner to PPD
. window packets . registration parameters	4. Calculate data retrieval envelopes for the window and store in various window packets.	. window packets
. input window packets	5. Check if any data is in the window. If not, set error flag.	
. error flags (diagnostic counters)	6. Check error flags. If errors, print message(s) and end.	Error messages
	7. Clear window number and reset page number.	. window number . page number
	8. If number of subwindows is not zero, generate subwindow maps (2.10.3.2).	Subwindow Maps
. symbol priority table . symbol table	9. If number of subwindows is zero, generate registered window map (2.10.3.3).	Map

Author:

Date: 02/14/79

Diagram ID: 2.10.3.2

Name: SUBWIN

Description: GENERATE REGISTERED SUBWINDOW MAPS

Input

. output window packet (KSYOWW)

Process

1. Save name of coordinate system the origin was specified in.
2. Save name of coordinate system the window vertices were specified in.
3. Store ticks coordinate system name in output window packet as origin coordinate system name and vertices coordinate system name.
4. Save window vertices in ticks coordinate system window packet (RLOWW).
5. Store tick interval as a window vertex in ticks coordinate system window packet (RLOWW). That tick will define a window, the other vertex being the origin, which will be different for each sub-window.
6. Compute on envelope in tick units that completely surrounds the window.

Output

. temporary origin coordinate system name

. temporary vertices coordinate system name

. output window packet (KSYOWW)

. temporary window vertices

. ticks coordinate system window packet (RLOWW)

Author: ______ Date: 02/16/79

Diagram ID: 2.10.3.2 Name: SUBWIN Description: GENERATE REGISTERED SUBWINDOW MAPS

Input	Process	Output
. number of subwindows →	7. Shrink that envelope by small increments until enough of the edge subwindows drop out that you can provide the user with not less than his requested number of subwindows.*	
. tick envelope dimensions →	8. Save the envelope dimensions. →	. tick envelope dimensions
	9. Generate origin for next subwindow within envelope. →	. origin
	10. Calculate data retrieval envelopes for the subwindow (RLOWW), transform to other coordinate systems, and store in remaining other window packets. →	. window packets
. input window packets →	11. Check to see if there is data in the window. If not, print message and skip to step 14.	Error messages
. output device →	12. Call appropriate registration and mapping routine depending on output device: a. If printer, generate registered map for subwindow (2.10.3.3).	

*Since several may drop out at once, it may be necessary to include more subwindows than he asked for in the envelope, in which case arbitrarily the last few may not be generated. The ones that drop out first will be the ones with the smallest portion of window in them.

Author:

Date: 02/16/79

Diagram ID: 2.10.3.2 Name: SUBWIN Description: GENERATE REGISTERED SUBWINDOW MAPS

Input	Process	Output
	b. If plotter, generate plotted map for subwindow (2.11.3.3).	
	c. If film, generate registered display file on tape for subwindow (2.12.3.3).	Registered Subwindow Maps
. number of subwindows	13. Subtract 1 from number of subwindows.	
	14. Repeat from step 8 until requested number of subwindows has been generated.	
. temporary origin coordinate system name . temporary vertices coordinate system name . temporary window vertices	15. Restore origin and vertices coordinate system names, and vertices of RLOWW.	

Author: ______

Diagram ID: 2.10.3.3 Name: MAPRNT

Date: 02/09/79

Description: GENERATE REGISTERED WINDOW MAP

Input

. origin

. tick table

. headings

. scanner window packet

Detection File

Process

1. Print origin.
2. Print tabular data (2.10.3.3.1).
3. Break window into sections for printing.
4. Get first tick in tick table.
5. Print headings for each print unit, print column scale (2.10.3.3.2), and print top border (2.10.3.3.3).
6. Compute first and last detection samples for data retrieval from Detection File.
7. Read detection line.
8. If window has more than two vertices, mark portions of detection line outside window but inside retrieval envelope as no data.

Output

Map:

- Tabular data

. current tick

Map:

- Headings
- Scale
- Top border

Author:

Diagram ID: 2.10.3.3   Name: MAPRNT

Date: 02/09/79

Description: GENERATE REGISTERED WINDOW MAP

Input	Process	Output
. file type . density range	9. If file type is density, screen the pixels for water, non-water, or no data as follows until all pixels in the line have been processed: a. Locate unpacking parameters of first density pixel for screening. b. Unpack pixel and determine type of pixel. c. If pixel is outside density range or is no data pixel, skip to step e, otherwise make entry in local frequency table. d. If water pixel: (1) The pixel maps to a particular PPD column. Calculate that column. (2) Add 1 to the count of water pixels there.	. output buffer for one line of output

Author: Date: 02/09/79

Diagram ID: 2.10.3.3 Name: MAPRNT Description: GENERATE REGISTERED WINDOW MAP

Input

- density range
- file type
- class range
- radiance range

Process

(3) Continue scanning water pixels and water words, storing counts, and making entries in local frequency table until a non-water or no data pixel is found.

e. If no data pixel, repeat from step b.

f. If non-water pixel, repeat from step b.

10. If file type is RADIANCE or CLASS:

a. Fill output buffer with non-priority flags.

b. Locate unpacking parameters of first detection pixel.

c. Unpack pixel.

d. If pixel's value is outside appropriate range or is no data pixel, skip to step i.

e. Each pixel in the line maps to a particular PPD column. Calculate that column for unpacked pixel.

Output

Author:

Date: 02/09/79

Diagram ID: 2.10.3.3 Name: MAPRNT Description: GENERATE REGISTERED WINDOW MAP

Input	Process	Output
	f. If column in output buffer currently has a non-priority flag, store current pixel value there.	. output buffer for one line of output
. symbol priority table . symbol table	g. If column output buffer currently does not have a non-priority flag and current pixel has a higher priority than that in the column, store that pixel value there.	
	h. Make entry in local frequency table.	
	i. Repeat from step c until all pixels have been processed.	
	11. Increment scan line number	
	12. Print map line (2.10.3.3.4)	Map: - Data lines
	13. Repeat from step 6 for every line in the window.	
	14. Print bottom border for each print unit (2.10.3.3.3), and print PPD column scale (2.10.3.3.2) for each unit.	- Border - Scale

Author:

Date: 02/09/79

Diagram ID: 2.10.3.3

Name: MAPRNT

Description: GENERATE REGISTERED WINDOW MAP

Input	Process	Output
	15. Repeat from step 4 for each section in the window. 16. Store frequency table as global for later use by TABULATE.	. frequency table

Author:

Date: 02/02/79

Diagram ID: 2.10.3.3.1 Name: NITHUG Description: PRINT TABULAR DATA

Input

. unit number to send tabulation to

. first two lines of heading

. program date

. program time

. scene ID

. classification parameters

. window number

. registration parameters

Process

1. Print heading.

2. Print date and time.

3. Print scene ID, classification summary, map window identification, and control summary.

Output

Tabular data:

- Map Heading
- Date
- Time
- Scene ID
- Classification summary
- Window Identification
- Control summary

Author:

Date: 02/02/79

Diagram ID: 2.10.3.3.2 Name: SAMSCL Description: PRINT SAMPLE SCALE

Input	Process	Output
. first sample in window . last sample in window . scale	1. Print scale of print columns or samples. 2. Repeat for every unit in the current section.	Map: - Sample scale printout

Author:

Date: 02/02/79

Diagram ID: 2.10.3.3.3 Name: BORDER Description: PRINT HORIZONTAL NEAT LINES

Input

. print buffer size

. print buffer

Process

1. Fill the print buffer with colons.
2. Print buffer.
3. Repeat for every unit in current section.

Output

Map:

- Horizontal neat lines

Author: ______ Date: 02/09/79

Diagram ID: 2.10.3.3.4 Name: LINOUT Description: PRINT MAP FILE

Input	Process	Output
. output buffer for one line of output →	1. Screen each pixel in line through symbol table as follows, until all pixels in line have been processed:	
	a. If non-priority flag, store no data symbol in corresponding map line buffer location.	
. symbol table →	b. If not non-priority flag, screen pixel through the symbol table and store symbol in corresponding map line buffer location.	
	2. Flag line that does not have any data due to scaling.	
. current tick →	3. Check if tick belongs in current line. If so:	
	a. If primary tick, insert in map line buffer.	
. symbol priority table →	b. If secondary tick, insert only if it will not be replacing a higher priority symbol.	
. tick table →	c. Get next tick in tick table.	
	d. Repeat from step 3.	

Author: ______

Diagram ID: 2.10.3.3.4 Name: LINOUT Date: 02/09/79 Description: PRINT MAP LINE

Input	Process	Output
	4. Print the line. 5. Repeat from step 1 for each unit in the current section. 6. Print any overprint characters. 7. Increment the print line number. 8. Reinitialize line.	Map: - One line of a map for each unit in the current section

Author: ______

Date: 02/02/79

Diagram ID: 2.10.4 Name: PRCEXI Description: EXIT FROM PRTCLASS

Input

. data/checkout mode flag (MDATAC)

. fatal error flag (NDFATL)

Process

1. If not in data/checkout mode, close alternate print files. If in data/checkout mode there are no files to close.

2. If fatal error occurred, exit in error.

3. If user does not want Detection Files saved, then delete Detection Files.

4. Exit.

Output

Author:

Date: 02/28/79

Diagram ID: 2.11 Name: PLTCLASS Description: PROGRAM PLOT REGISTERED MAPS

Input	Process	Output
Detection File Registration Parameter File →	1. Initialize by opening files, loading system date and time, printing program start message and setting mode switches (2.11.1).	
Card images: → - User directives	2. Read commands and specifications, and save command name. →	. command name
	3. If command defines or selects specifications, store them in applicable variables (2.11.2). →	. specification variables
	4. If command is AREA, calculate and print area within user-specified closed figure (2.13.4). →	Area
	5. If command is LENGTH, calculate and print length of user-specified line segment or series of connected line segments (2.13.5). →	Length
	6. If command is MAP, plot registered map(s) (2.11.3). →	Maps
	7. If command is TABULATE, print tabulation of frequency (2.7.5). →	Tabulations
	8. If command is HISTOGRAM, plot histogram of frequency (2.7.5). →	Histograms

Author:

Date: 02/28/79

Diagram ID: 2.11 Name: PLTCLASS Description: PROGRAM PLOT REGISTERED MAPS

Input	Process	Output
	9. If command is EXIT, exit from PLTCLASS (2.11.4). 10. Print message if illegal command. 11. Repeat from step 2.	Error message

Author:

Date: 02/28/79

Diagram ID: 2.11.1   Name: PLCXQT   Description: INITIALIZE

Input	Process	Output
Detection Files Registration Parameter File	1. Get system date, time, and mode switches. Print program start message and place message in log file. 2. Open Detection File and identify file types. 3. Print Landsat scene ID and load registration parameters. 4. Initialize default commands.	Program start message . mode switches . system date . system time Log File . file types . input window packet Scene ID . registration parameters

Author: ______ Date: 02/28/79

Diagram ID: 2.11.2 Name: ______ Description: PROCESS COMMANDS THAT SPECIFY

Input	Process	Output
. command name	1. If command is CLASS, store class range to be used in processing windows (2.10.2.6).	. class range
	2. If command is DENSITY, store density range to be used in processing windows (2.10.2.1).	. density range
	3. If command is HEADING, store text to be used in window headings (2.10.2.2).	. heading lines
	4. If command is MERIDIAN, store non-standard UTM central meridian (2.10.2.3).	. UTM central meridian
	5. If command is RADIANCE, store radiance range to be used in processing windows (2.7.2.6).	. radiance range
	6. If command is SCALE, store map scale desired (2.10.2.4).	. scale
	7. If command is SYMBOL, store symbol in appropriate bins, depending on value or range specified (2.6.2.3).	. symbol table
	8. If command is TICK, store tick interval (2.7.2.8).	. tick interval

Author: ______

Date: 02/28/79

Diagram ID: 2.11.2 Name: ______

Description: PROCESS COMMANDS THAT SPECIFY

Input	Process	Output
	9. If command is WINDOW, store window vertices (2.7.2.9).	. window vertices
	10. If command is ZONE, store UTM zone and standard UTM central meridian (2.7.2.10).	. zone . UTM central meridian
	11. If command is PLOTTER, store plotter characteristics (2.11.2.1).	. plotter characteristics
	12. If illegal command, print error message.	Error message

Author:

Date: 02/20/79

Diagram ID: 2.11.2.1 Name: KMDPLO Description: STORE PLOTTER CHARACTERISTICS

Input	Process	Output
Card image	1. Check plot sheet size and units. If invalid, set error flag.	. sheet size . units
	2. Check device ID. If invalid, set error flag.	. device ID (TAPE or remote site ID)
	3. Check plotter model. If invalid, set error flag.	. model (Calcomp or Gerber number)
	4. Check number of pens. If invalid, set error flag.	. number of pens
. confirm switch	5. If confirm switch is on, print sheet size, device ID, model, and number of pens.	Sheet size Device ID Model Number of pens
	6. Blank command word.	. command word

Author: ______ Date: 02/20/79

Diagram ID: 2.11.3 Name: PLCMAP Description: PLOT REGISTERED MAPS

Input	Process	Output
Card image	1. Get/check the number of subwindows. If not valid, set error flag.	. number of subwindows . error flag
. symbol table	2. Complete the symbol table by checking validity of requested symbols, and flagging those symbols that require a special character (2.11.3.1).	. symbol table
. scale	3. Calculate mapping equations for scale.	. transformation coefficients from corrected scanner to PPD
. window packets . registration parameters	4. Calculate data retrieval envelopes for the window and store in various window packets.	. window packets
. input window packets	5. Check if any data is in the window. If not, set error flag.	
. error flags (diagnostic counters)	6. Check error flags. If errors, print message(s) and return.	Error messages
	7. Clear window number and reset page number	. window number . page number
	8. If number of subwindows is not zero, plot subwindow maps (2.10.3.2).	Subwindow map plots
	9. If number of subwindows is zero, plot window map (2.11.3.2).	Window map plot

Author: ____________

Date: 02/20/79

Diagram ID: 2.11.3.1 Name: SYMTPL Description: COMPLETE SYMBOL TABLE

Input	Process	Output
.symbol table	1. Check validity of each multicharacter symbol. If not DOT, DAS[H], LIN[E], TRI[ANGLE], or SQU[ARE], print error message(s) and return.	Error messages
	2. If multicharacter symbol is one of the above five, store appropriate value (one through five) in that symbol table entry, in the area usually used for the number of overprint characters. It will be used to indicate the appropriate special-character generating routine that generates that symbol.	.symbol table

Author:

Diagram ID: 2.11.3.2 Name: MAPLOT Date: 02/28/79

Description: PLOT WINDOW MAP

Input

. origin

. scene ID
. window number
. registration parameters
. classification parameters

Process

1. Plot origin information.
2. Plot map heading and tabular data (2.11.3.2.1).
3. Break window into sections for plotting.
4. Plot summary information.
5. Plot neat lines (2.11.3.2.2).
6. Plot marginal ticks (2.11.3.2.3).
7. Plot pixels and generate local frequency table (2.11.3.2.6).
8. Plot internal ticks (2.11.3.2.4).
9. Plot no data areas (2.11.3.2.5).
10. Repeat from step 4 for each section in window.
11. Store frequency table as global for later use by TABULATE.

Output

Map plot:
- Origin
- Headings
- Tabular data
- Summary information
- Neat lines
- Marginal ticks
- Pixels
- Internal ticks
- No data areas

. frequency table

Author: ______ Date: 02/28/79

Diagram ID: 2.11.3.2.1 Name: ______ Description: PLOT TABULAR DATA

Input	Process	Output
. heading lines . system date . system time . scene ID . classification parameters . window number . registration parameters	1. Plot heading lines. 2. Plot date and time. 3. Plot scene ID, classification parameters, window number, and registration parameters.	Tabular data: - Heading lines - Date and time - Scene ID - Classification parameters - Window number - Registration parameters

Author: ______

Date: 02/28/79

Diagram ID: 2.11.3.2.2   Name: NTLNPL   Description: PLOT NEAT LINES

Input	Process	Output
. PPD envelope dimensions	1. Plot neat lines surrounding the window.	Map plot: - Neat lines

C4

Author:

Date: 03/01/79

Diagram ID: 2.11.3.2.3 Name: MARGIN Description: PLOT MARGINAL TICKS

Input

. primary tick interval

. secondary tick interval

. number of pens

. output window packet (containing primary and secondary tick coordinate system names)

Process

1. Calculate PPD position of first primary tick on left axis.
2. If number of pens is greater than 1, change to second pen color.
3. Using primary tick interval, plot each tick mark along the left axis, oriented directionally with the primary tick coordinate system.
4. Calculate PPD position of first primary tick on bottom axis.
5. Using primary tick interval, plot each tick mark along the bottom axis, oriented directionally with the primary tick coordinate system.
6. Calculate PPD position of first secondary tick on top axis.
7. If number of pens is greater than 2, change to third pen color.

Output

Map plot:

- Left marginal ticks

- Bottom marginal ticks

Author: ____________ Date: 03/01/79

Diagram ID: 2.11.3.2.3 Name: MARGIN Description: PLOT MARGINAL TICKS

Input	Process	Output
	8. Using secondary tick interval, plot each tick mark along the top axis, oriented directionally with the secondary tick coordinate system.	Map plot: - Top marginal ticks
	9. Calculate PPD position of first secondary tick on right axis.	
	10. Using secondary tick interval, plot each tick mark along the right axis, oriented directionally with the secondary tick coordinate system.	- Right marginal ticks
	11. Return to pen color of first pen.	

Author: ____________

Diagram ID: 2.11.3.2.4 Name: ____________

Date: 02/28/79

Description: PLOT INTERNAL TICK MARKS

Input

- . number of pens
- . primary ticks coordinate system name
- . primary tick interval
- . scale
- . scanner envelope
- . UTM envelope
- . GED envelope
- . secondary ticks coordinate system name
- . secondary tick interval

Process

1. If number of pens is greater than 1, change to second pen color.
2. Set ticks coordinate system name equal to primary ticks coordinate system name, and tick interval equal to primary tick interval.
3. Calculate tick orientation and scale, depending on ticks coordinate system name, and scale.
4. Plot tick at proper orientation and scale.
5. Repeat from step 4 for each tick in envelope.
6. If number of pens is greater than 2, change to third pen color.
7. Set ticks coordinate system name equal to secondary ticks coordinate system name, and tick interval equal to secondary tick interval.
8. Repeat steps 3, 4, and 5 for secondary ticks.
9. Restore pen color to first pen.

Output

Map plot:

- Primary and secondary internal tick marks

Author: ______

Date: 02/28/79

Diagram ID: 2.11.3.2.5

Name: NODAPL

Description: PLOT NO DATA AREAS

Input

. window vertices

. envelope dimensions

Process

1. Calculate, in PPD coordinates, a series of rectangles that cover all no data areas in PPD envelope.
2. For each rectangle delineate boundaries of rectangle and plot a series of diagonal line segments an appropriate distance apart to fill the rectangle.

Output

No data areas:

- Boundaries
- Diagonal lines

Author: ______

Date: 02/28/79

Diagram ID: 2.11.3.2.6 Name: PLTPIX Description: PLOT PIXELS

Input

. scanner output window packet

Detection File

. density range
. class range
. radiance range
. file type

Process

1. Compute first and last detection samples for data retrieval from Detection File.
2. Read detection line.
3. If window has more than two vertices, mark portions of detection line outside window but inside envelope as no data.
4. Get pixel.
5. If pixel is not in appropriate range for this file type, set symbol as no data and skip to step 11.
6. If pixel is no data pixel, skip to step 11.
7. Add 1 to local count in frequency table.
8. Calculate PPD location to map that first pixel to.
9. If file type is density and pixel is non-water, set symbol to no data and skip to step 11.
10. Find symbol to be used for that pixel from symbol table.

Output

Author:

Date: 02/28/79

Diagram ID: 2.11.3.2.6 Name: PLTPIX Description: PLOT PIXELS

Input	Process	Output
	11. Continue scanning pixels until a pixel represented by a different symbol is found. 12. If pixel string is no data symbols, skip to step 16. 13. Calculate PPD location to map the last pixel in string to, and number of pixels found. 14. If symbol for the pixel string is not LINE: a. Calculate PPD location for the pixel. b. Plot pixel. c. Repeat from step a for each pixel in string. 15. If symbol for this pixel string is LINE, plot vector. 16. Repeat from step 4 until all pixels in line have been processed. 17. Increment the scan line number.	Map plot: - Pixels

Author:

Date: 02/28/79

Diagram ID: 2.11.3.2.6 Name: PLTPIX Description: PLOT PIXELS

Input	Process	Output
	18. Repeat from step 1 for every line in the window.	

Author:

Diagram ID: 2.11.4 Name: PLCEXI Date: 02/28/79

Description: EXIT PLTCLASS

Input	Process	Output
.fatal error flag (NDFATL) →	1. If fatal error occurred, exit in error.	
Detection Files →	2. If user does not want Detection Files saved, delete Detection Files.	
	3. Exit.	

Author:

Date: 02/16/79

Diagram ID: 2.12 Name: FLMCLASS Description: PROGRAM CREATE FILM OUTPUT

Input	Process	Output
Detection Files Registration Parameter File	1. Initialize by opening files, loading system date and time, identifying program name, and setting mode switches (2.12.1).	Program start message Log File . system date . system time . mode switches
Card images: - User directives	2. Read commands and specifications.	. command name
	3. If command defines or selects specifications, store them in applicable variables (2.12.2).	. specification variables
	4. If command is AREA, calculate and print area within user-specified closed figure (2.13.4).	Area
	5. If command is LENGTH, calculate and print length of user-specified line segment or series of connected line segments (2.13.5).	Length
	6. If command is MAP, generate color film registered display by building an image data format tape which will be sent to the Production Film Converter (PFC) (2.12.3).	JSC Universal Image Data Format tape to the PFC
	7. If command is TABULATE, print tabulation of frequency (2.6.4).	Tabulations

Author:

Diagram ID: 2.12 Name: FLMCLASS

Date: 02/16/79

Description: PROGRAM CREATE FILM OUTPUT

Input	Process	Output
	8. If command is HISTOGRAM, print histogram of frequency (2.7.5).	Histogram
	9. If command is EXIT, exit from FLMCLASS (2.12.4).	
	10. Print message if illegal command.	Error message
	11. Repeat from step 2.	

Author:

Diagram ID: 2.12.1 Name: FLMXQT Date: 02/07/79 Description: INITIALIZE

Input	Process	Output
Detection Files	1. Get system date, time, and mode switches. Identify program name to be placed as a message in the log file.	Program start message
Registration Parameter File	2. Open Detection File and identify file types.	Scene identification
	3. Identify Landsat scene and load registration parameters.	Log File
	4. Initialize default commands.	. mode switches
		. system date
		. system time
		. file types
		. input window packets
		. registration parameters

Author:

Date: 02/16/79

Diagram ID: 2.12.2

Name:

Description: PROCESS COMMANDS THAT SPECIFY

Input	Process	Output
. command name Card image	1. If command is DENSITY, store density range to be used in processing windows (2.10.2.1).	. density range
	2. If command is CLASS, store class range to be used in processing windows (2.10.2.6).	. class range
	3. If command is HEADING, store text to be used in window headings (2.10.2.2).	. headings
	4. If command is MERIDIAN, store non-standard UTM central meridian (2.10.2.3).	. UTM central meridian
	5. If command is ORIGIN, store window origin (2.7.2.5).	. origin
	6. If command is RADIANCE, store radiance range to be used in processing windows (2.7.2.6).	. radiance range
	7. If command is SCALE, store map scale desired (2.10.2.4).	. scale
	8. If command is COLOR, store color in appropriate bins in symbol table, according to value or range specified (2.7.2.3).	. color symbol table

Author:

Diagram ID: 2.12.2 Name:

Date: 02/16/79

Description: PROCESS COMMANDS THAT SPECIFY

Input

Process

9. If command is TICK, store tick interval (2.7.2.8).
10. If command is WINDOW, store window vertices (2.7.2.9).
11. If command is ZONE, store UTM zone and standard central meridian (2.10.2.10).
12. If illegal command, print message and return.

Output

. tick interval

. window vertices

. zone

. UTM central meridian

Error message

Author: ______

Date: 02/13/79

Diagram ID: 2.12.3 Name: FLMMAP Description: GENERATE COLOR FILM REGISTERED DISPLAY

Input	Process	Output
Card image	1. Get/check the number of subwindows. If not valid, set error flag.	. number of subwindows . error flag
. color symbol table	2. Since film emulsion does not respond in the same manner as the RAMTEK to the same intensities of color guns, apply a color transformation function to the color symbol table, making color film output appear the same as RAMTEK color output (2.12.3.1).	. color symbol table
. scale	3. Calculate mapping equations for scale.	. transformation coefficients from corrected scanner to PPD
. window packets . registration parameters	4. Calculate data retrieval envelopes for the window and store in various window packets.	. window packets
. input window packets	5. Check if any data is in the window. If not, set error flag.	
. error flags (diagnostic counters)	6. Check error flags. If errors, print message(s) and return.	Error messages
	7. Clear window number.	. window number

Author: ______ Date: 02/13/79

Diagram ID: 2.12.3 Name: FLMMAP Description: GENERATE COLOR FILM REGISTERED DISPLAY

Input	Process	Output
. symbol priority table . color symbol table Detection File	8. If number of subwindows is not zero, generate registered display file on tape for each subwindow (2.10.3.2). 9. If number of subwindows is zero, register pixels and generate registered display file on tape for window (2.12.3.2).	JSC Image Data Format tape - Display file(s)

Author:

Diagram ID: 2.12.3.1   Name:

Date: 02/13/79

Description: APPLY COLOR TRANSFORMATION FUNCTION TO SYMBOL TABLE

Input

- color symbol table

Process

1. Apply transformation to color gun 1's intensity for each class or radiance range in the symbol table.
2. Apply transformation to color gun 2's intensity for each class or radiance range in the symbol table.
3. Apply transformation to color gun 3's intensity for each class or radiance range in the symbol table.

Output

- color symbol table

Author: ____

Diagram ID: 2.12.3.2 Name: FLMPRT

Date: 02/16/79

Description: REGISTER PIXELS AND GENERATE REGISTERED DISPLAY FILE ON TAPE

Input	Process	Output
. PPD window packet	1. Crop output window to fit maximum production film converter size.	. PPD window packet
	2. Build imagery header record (2.12.3.3.1).	Tape File: - Imagery header record
. tick table	3. Get first tick in tick table.	
	4. Compute first and last detection samples for data retrieval from Detection File.	
Detection File	5. Read detection line.	
	6. If window has more than two vertices, mark portions of detection line outside window but inside envelope as no data.	
. file type	7. If file type is DENSITY, screen the pixels for water, non-water, or no data as follows until all pixels in line have been processed: a. Determine unpacking parameter of first density pixel for screening.	

Author: ______ Date: 02/16/79

Diagram ID: 2.12.3.2 Name: FLMPRT Description: REGISTER PIXELS AND GENERATE REGISTERED DISPLAY FILE ON TAPE

Input	Process	Output
. density range	b. Unpack pixel and determine pixel type. c. If pixel is outside density range or is no data, skip to step e, otherwise make entry in local frequency table. d. If water pixel: (1) The pixel maps to a particular PPD column. Calculate that column. (2) Add 1 to count of water pixels there. (3) Continue scanning water pixels and water words, storing counts, and making entries in local frequency table until a non-water or no data pixel is found. e. If no data pixel repeat from step b. f. If non-water pixel, repeat from step b.	. output buffer for one line of output

Author:

Date: 02/16/79

Diagram ID: 2.12.3.2   Name: FLMPRT   Description: REGISTER PIXELS AND GENERATE REGISTERED DISPLAY FILE ON TAPE

Input	Process	Output
. file type	8. If file type is RADIANCE or CLASS:	
	a. Locate unpacking parameters of first detection pixel.	
	b. Fill output buffer with non-priority flags.	
. radiance range . class range	c. Unpack pixel. If outside of appropriate range, skip to step h. If no data value, skip to step h.	
	d. Each pixel in the line maps to a particular PPD column. Calculate that column for unpacked pixel.	
	e. If column in output buffer currently has a non-priority flag, store current pixel value there.	. output buffer for one line of output
. symbol table . symbol priority table	f. If column in output buffer does not have a non-priority flag and the pixel has a higher priority than that in the column, store that pixel value there.	

Date: 02/16/79

Author:

Diagram ID: 2.12.3.2 Name: FLMPRT Description: REGISTER PIXELS AND GENERATE REGISTERED DISPLAY FILE ON TAPE

Input

Process

g. Make an entry in the local frequency table.

h. Repeat from step c until all pixels in line have been processed.

9. Increment the scan line number.

10. Write output buffer to tape (2.12.3.3.2).

11. Repeat from step 4 for every line in the window.

12. Store frequency table as global for later use by TABULATE.

Output

Tape File:

- Data sets (scan lines)

. frequency table

Author:

Date: 02/08/79

Diagram ID: 2.12.3.2.1 Name: FLMHDR Description: BUILD IMAGERY HEADER RECORD

Input	Process	Output
. window number →	1. If this is first window processed, rewind output tape.	
. system date →	2. Store system ID and date in header buffer.	
. tape sequence number →	3. Store tape sequence number (1 if not initialized), number of active channels (3), and constants.	
. number of PPD columns in window (the record lengths are based on the number of output pixels per scan line) →	4. Calculate and store logical record lengths of ancillary data for each scan line, and image data for each scan line. →	. header buffer
. PPD window packet →	5. Store PPD column number for first and last pixel in scan line.	
. GED window packet →	6. Store latitude and longitude of first pixel in first scan line.	
	7. Store $\frac{\text{PPD columns in window}}{1000}$ (thousands of scan lines per frame).	

Author:

Date: 02/14/79

Diagram ID: 2.12.3.2.1

Name: FLMHDR

Description: BUILD IMAGERY HEADER RECORD

Input

. altitude in meters
. satellite yaw
. first heading line
. window number

Process

8. Store satellite altitude and yaw.
9. Store window title.
10. Write header buffer to tape. If end-of-tape occurs while writing:
    a. If first window, bad tape. Rewind and remove old tape; request, mount, and rewind new tape, and repeat from step 10 once.
    b. If not first window, back up to last window on tape and put 3 EOF's after the window. Rewind and remove tape, mount and rewind new tape, and repeat from step 10.
11. Save header buffer on disk in case of end-of-tape while writing data sets (scan lines).

Output

. header buffer

Tape File:

- Imagery header record

Header Buffer File

Author: ______ Date: 02/08/79

Diagram ID: 2.12.3.2.2 Name: FLMOUT Description: WRITE SCAN LINE BUFFER TO TAPE

Input	Process	Output
. current PPD line number	1. Build ancillary data record including current line number.	
	2. Screen each pixel in line through color symbol table as follows, until all pixels in line have been processed:	
	a. If non-priority flag, store intensities of 3 color guns indicating background color for that pixel in corresponding image data buffer location.	. image data buffer
. color symbol table	b. If not non-priority flag, screen pixel through color symbol table and store intensities of 3 color guns for that pixel in corresponding image data buffer location.	
	c. Repeat for each pixel in line.	
	3. Flag line that does not have any data due to scaling.	
	4. Check if tick belongs in current line. If so:	

Author:

Date: 02/08/79

Diagram ID: 2.12.3.2.2 Name: FLMOUT Description: WRITE SCAN LINE BUFFER TO TAPE

Input

. tick table

. Header Buffer File

. tape sequence number

Process

a. Insert tick mark in place.

b. Get next tick in tick table.

c. Repeat from step 4.

5. If not first window line, (data set), back up two EOF marks leaving one EOF at end of last data set.

6. Write image data buffer to tape, and 3 EOF's after. If end-of-tape occurs while writing:

   a. Increment tape sequence number by 1.

   b. Back up and erase last of 3 preceding EOF marks after last data set, leaving two.

   c. Rewind and remove output tape.

   d. Mount and rewind new output tape.

   e. Retrieve stored header buffer.

   f. Replace tape sequence number on local header buffer.

Output

Tape File:

- Imagery data set (scan line)

Author: ______ Date: 02/12/79

Diagram ID: 2.12.3.2.2 Name: FLMOUT Description: WRITE SCAN LINE BUFFER TO TAPE

Input	Process	Output
	g. Write new header buffer to tape. h. Repeat from step 7 once. 7. Increment PPD line number. 8. Reinitialize image data buffer.	

Author:

Date: 02/06/79

Diagram ID: 2.12.4 Name: FLMEXI Description: EXIT FROM FLMCLASS

Input	Process	Output
. fatal error flag →	1. If fatal error flag, exit in error.	
Detection Files →	2. If user does not want Detection Files saved, delete Detection Files.	
	3. Exit.	

Author: ____________ Date: 02/16/79

Diagram ID: 2.13 Name: CRTCLASS Description: PROGRAM PRODUCE COLOR CRT REGISTERED DISPLAYS

Input	Process	Output
Detection Files Registration Parameter File	1. Initialize by opening files, loading system date and time, printing program start message, and setting mode switches (2.13.1).	Program start message . system date . system time . mode switches
	2. Read commands and specifications and save command name.	. command name
	3. If command defines or selects specifications, store them in applicable variables (2.13.2).	. specification variables
	4. If command is AREA, calculate and print area within user-specified closed figure (2.13.4).	Area
	5. If command is LENGTH, calculate and print length of user-specified line segment or series of connected line segments (2.13.5).	Length
	6. If com[illegible]nd [illegible]AP, ge[illegible] color registered disp[illegible] (2.13.3).	Color Registered Displays on CRT
	7. If command is HISTOGRAM, display histogram of frequency (2.7.5).	Histogram

Author:

Date: 02/16/79

Diagram ID: 2.13

Name: CRTCLASS

Description: PROGRAM PRODUCE COLOR CRT REGISTERED DISPLAYS

Input	Process	Output
	8. If command is TABULATE, print tabulation of frequency (2.7.4).	Tabulation
	9. If command is EXIT, terminate (2.13.6).	
	10. Print message if illegal command.	Error messages
	11. Repeat steps 2 through 8.	

Author:

Date: 02/07/79

Diagram ID: 2.13.1 Name: CRTXQT Description: INITIALIZE

Input	Process	Output
	1. Get system date, time, and mode switches. Print program start message and place a message in the log file.	Program start message . mode switches . system date . system time
Detection File	2. Open Detection File and identify file types.	Log File . file types . input window packets
Registration Parameter File	3. Print Landsat scene ID and load registration parameters.	Scene identifcation . registration parameters
	4. Initialize default commands.	

Author: ____________ Date: 02/16/79

Diagram ID: 2.13.2 Name: ____________ Description: PROCESS COMMANDS THAT SPECIFY

Input	Process	Output
. command name Card image	1. If command is CLASS, store class range to be used in processing windows (2.10.2.6).	. class range
	2. If command is COLOR, store color and value or range (2.7.2.3).	. color symbol table
	3. If command is DENSITY, store density range to be used in processing windows (2.10.2.1).	. density range
	4. If command is HEADING, store text to be used in window headings (2.10.2.2).	. headings
	5. If command is MERIDIAN, store non-standard UTM central meridian (2.10.2.3).	. meridian
	6. If command is ORIGIN, store window origin (2.7.2.5).	. origin
	7. If command is RADIANCE, store radiance range to be used in processing windows (2.7.2.6).	. radiance range
	8. If command is SCALE, store map scale desired (2.10.2.4).	. scale
	9. If command is TICK, store tick interval (2.7.2.8).	. tick interval

Author:

Date: 02/16/79

Diagram ID: 2.13.2 Name:

Description: PROCESS COMMANDS THAT SPECIFY

Input	Process	Output
	10. If command is WINDOW, store window vertices (2.7.2.9).	. window vertices
	11. If command is ZONE, store UTM zone and standard UTM central meridian (2.7.2.10).	. zone . UTM central meridian
	12. If illegal command, print message.	Error message

Author:

Date: 02/08/79

Diagram ID: 2.13.3 Name: CRTMAP Description: GENERATE COLOR REGISTERED DISPLAY

Input	Process	Output
. scale	1. Calculate mapping equations for scale.	. transformation coefficients from corrected scanner to printer/plotter device (PPD) coordinates.
. window packets . registration parameters	2. Calculate data retrieval envelopes for the window and store in various window packets.	. window packets
. input window packets	3. Check if any data is in the window. If not, set error flag.	. error flag
. error flags (diagnostic counters)	4. Check error flags. If errors, print message(s) and return.	Error messages
	5. Clear window number.	. window number
Detection File	6. Register pixels and display (2.13.3.1).	Registered Display

Author: ______

Date: 02/12/79

Diagram ID: 2.13.3.1 Name: CRTPRT Description: REGISTER PIXELS AND DISPLAY

Input	Process	Output
. PPD window packets →	1. Crop output window to fit 512 printer/plotter device (PPD) columns by 512 PPD lines.	
. scanner window →	2. Compute first and last detection samples for data retrieval from Detection File.	
Detection File →	3. Read detection line.	
	4. If window has more than two vertices, mark portions of detection line outside window but inside retrieval envelope as no data.	
. file type →	5. If file type is DENSITY screen the detection line for water, non-water, or no data as follows until all pixels are processed:	
	a. Locate unpacking parameters of first density pixel for screening.	
	b. Unpack pixel and find pixel type.	
. density range →	c. If pixel is outside density range or is no data pixel, skip to step e, otherwise make entry in local frequency table.	

Author:

Date: 02/07/79

Diagram ID: 2.13.3.1 Name: CRTPRT Description: REGISTER PIXELS AND DISPLAY

Input	Process	Output
. density range	d. If water pixel: (1) The pixel maps to a particular PPD column. Calculate that column. (2) Add 1 to the count of water pixels there. (3) Continue scanning water pixels and water words, storing counts, and making entries in local frequency table until a non-water or no data pixel is found. e. If no data pixel, repeat from step b. f. If non-water pixel, repeat from step b. 6. If file type is RADIANCE or CLASS: a. Locate unpacking parameters of first detection pixel. b. Fill output buffer with non-priority flags.	. PPD scan line buffer

Author:

Date: 02/07/79

Diagram ID: 2.13.3.1 Name: CRTPRT Description: REGISTER PIXELS AND DISPLAY

Input

- file type
- class range
- radiance range
- symbol priority table
- color symbol table

Process

c. Unpack pixel.

d. If pixel is outside appropriate range or is a no data pixel, skip to step i.

e. Each pixel in the line maps to a particular PPD column. Calculate that column for unpacked pixel.

f. If column in output buffer currently has a non-priority flag, store current pixel value there.

g. If column in output buffer currently does not have a non-priority flag and the current pixel has a higher priority than that in the column, store that pixel value there.

h. Make entry in local frequency table.

i. Repeat from step c until all pixels in line have been processed.

7. Increment scan line number.

Output

- PPD scan line buffer

Author:

Diagram ID: 2.13.3.1 Name: CRTPRT

Date: 02/07/79

Description: REGISTER PIXELS AND DISPLAY

Input

- primary tick interval
- secondary tick interval
- window packets
- heading lines

Process

8. Display scan line on CRT (2.13.3.1.2).
9. Repeat from step 2 for every line in the window.
10. Display tabular data (2.13.3.1.1).
11. Generate and display ticks filling the window.
12. Display headings and user coordinate information on CRT.
13. Store frequency table as global for later use by TABULATE.

Output

Registered Display:
- Display
- Tabular data
- Ticks
- Headings
- User coordinate information

. frequency table

Author: ______ Date: 02/01/79

Diagram ID: 2.13.3.1.1 Name: ______ Description: DISPLAY TABULAR DATA

Input

. scene ID
. classification parameters
. window number
. registration parameters

→

Process

1. Display scene ID, classification summary, window number, and control summary.

→

Output

Tabular data:

- Scene ID
- Classification summary
- Window number
- Control summary

Author: ____________

Diagram ID: 2.13.3.1.2   Name: CRTOUT   Date: 03/01/79

Description: DISPLAY SCAN LINE ON CRT

Input	Process	Output
. PPD scan line buffer . color symbol table	1. Screen each pixel in line through symbol table as follows, until all pixels in line have been processed: a. If non-priority flag, store background color in corresponding reformatted scan line buffer location b. If not non-priority flag, screen pixel through color symbol table and store color in corresponding reformatted scan line buffer location. c. Repeat for each pixel in line. 2. Flag line that does not have any data due to scaling. 3. Display scan line on CRT. 4. Reinitialize PPD scan line buffer.	. reformatted scan line buffer Display: - Scan line . PPD scan line buffer

Author:

Date: 03/08/79

Diagram ID 2.13.4 Name: KMDARE Description: CALCULATE AREA OF CLOSED FIGURE

Input	Process	Output
Card image(s) . registration parameters . window packets	1. Get and check name of coordinate system the line segment vertices are specified in (MSA, GED, or UTM). If invalid, print message and set error flag. 2. If name of coordinate system is GED or UTM, and registration parameters are not available, print message and use nominal registration parameters. 3. Get and check line segment vertices. a. If outside window, print message and set error flag. b. If line segments are not contiguous, print message and set error flag. c. If figure is not closed, generate an extra set of vertices to close figure. 4. Get and check the units user wants area specified in. If not PIXELS or blanks, print message and set error flag.	Error messages . error flag

Author: ____________ Date: 03/08/79

Diagram ID: 2.13.4 Name: KMDARE Description: CALCULATE AREA OF CLOSED FIGURE

Input	Process	Output
. registration parameters	5. If error flag is on, return. 6. If line segment vertices are specified in GED coordinates, transform to scanner coordinates. 7. If line segment vertices are specified in UTM coordinates, transform to scanner coordinates. 8. Count the number of pixels in closed figure. 9. Print number of pixels.	Number of pixels

Author:

Date: 03/07/79

Diagram ID: 2.13.5 Name: KMDLEN Description: CALCULATE LINE SEGMENT LENGTH

Input	Process	Output
Card image(s) . registration parameters . window packets . UTM zone . UTM central meridian	1. Get and check name of coordinate system the line segment vertices are specified in (MSA, GED, or UTM). If invalid, print message and set error flag. 2. If name of coordinate system is GED or MSA (scanner) and registration parameters are not available, print message and use nominal registration parameters. 3. Get and check line segment vertices. a. If outside window, print message and set error flag. b. If line segment(s) are not contiguous, print message and set error flag. 4. Get and check the units user wants length specified in. If not METERS or KILOMETERS, print message and set error flag. 5. If UTM zone and central meridian not specified, print message and set error flag.	Error messages . error flag

Author: ____________ Date: 03/07/79

Diagram ID: 2.13.5 Name: KMDLEN Description: CALCULATE LINE SEGMENT LENGTH

Input	Process	Output
	6. If error flag is on, return. 7. If line segment vertices are specified in GED or MSA, transform to UTM. 8. Calculate change in UTM easting and northing in meters between two line segment vertices. 9. Calculate length of line segment in meters. 10. If user wants length specified in kilometers, divide length by 1000. 11. Add length to line length count. 12. Repeat from step 8 for each contiguous line segment specified. 13. Print line length count.	Length

Author:

Diagram ID: 2.13.6 Name: CRTEXI Date: 02/01/79 Description: EXIT FROM CRTCLASS

Input	Process	Output
· fatal error flag →	1. If fatal error occurred, exit in error.	
· Detection Files →	2. If user does not want Detection Files saved, delete Detection Files.	
	3. Exit CRTCLASS.	

Author: ______ Date: ______

Diagram ID: 2.14 Name: ANOMALIES Description: DELETE ANOMALIES FROM DETECTION FILE

Input

Detection File
- Class
- Radiance

Signature File
- Label table
- Symbol table

Radius Table File

Card images

Process

1. Open Detection File, Signature File, and Radius Table File and save radius table (2.14.1).
2. Build a spectral neighbor table from the label table and save.
3. Read and save command.
4. If command is RADIUS, get radius of closure for specified radiance range or class and build a radius table (2.14.3).
5. If command is ANOMLY, scan Detection File and process specified anomalies to output to a new Detection File (2.14.4).
6. If command is EXIT, terminate the job.
7. If invalid command, print a message.
8. Repeat steps 3 through 7.

Output

. type of Detection File

. radius table

. spectral neighbor table

. command name

Detection File (copy)

Error message

Author: ______

Date: ______

Diagram ID: 2.14.1

Name: ______

Description: INITIALIZE FILES FOR ANOMALY PROGRAM

Input

Detection File

Signature File
- Label table
- Symbol table

Radius Table File

Process

1. Open Detection File and read header record.
2. Print header information on print file.
3. Open Signature File.
4. Open Radius Table File; read the radius table and save.

Output

. Detection File information

. type of Detection File

Identification of Detection File

. radius table

Author: ______ Date: ______

Diagram ID: 2.14.3 Name: ______ Description: RADIUS

Input

Card image

Signature File
- Symbol table

. radius table

Process

1. Get class number.
2. If class is not a number, print a message and set error flag.
3. Get radius of closure, which is expressed in number of pixels.
4. If radius is not a number, print a message and set error flag.
5. If radius table does not exist, copy the class and low and high radiance values from the symbol table file into the radius table and initialize radius of closure to 0.
6. If class number from card is not present in radius table, print error message, set error flag.
7. If error occurred, return.
8. Insert radius of closure in radius table.

Output

Error message

. error flag

. radius table

Author: ______________________ Date: ____________

Diagram ID: 2.14.3.2 Name: ______________________ Description: CLASS ANOMALIES

Input

. line of anomaly

. sample of anomaly

Detection File

- Class

. radius table

Process

1. Using scan line and sample location of anomaly pixel, determine the class of the pixel in the Detection File.

2. Get radius of closure for this class from the radius table.

3. Determine the boundaries of the complete anomaly by locating all contiguous pixels of the same class.

4. Using the top boundary of the anomaly and the radius of closure, calculate the starting scan line for saving data outside the anomaly.

5. Reinitialize at the beginning of Detection File.

6. Read and output all scan lines (records) that will not change onto Detection File copy.

Output

. beginning line boundary

. ending line boundary

. beginning sample boundaries/line

. ending sample boundaries/line

. starting scan line

Detection File (copy)

Author: ______ Date: ______

Diagram ID: 2.14.3.2 Name: ______ Description: CLASS ANOMALIES

Input	Process	Output
. spectral neighbor table	7. Read into packed buffer, unpack pixel data, and store in unpacked buffer to provide enough surrounding pixels to equal the radius of closure in all directions for the current anomaly line. 8. By looking at surrounding pixels and using the spectral neighbor table, determine if any anomaly pixels in the current anomaly line should change their class. If a change is needed, determine the new class from classes of the surrounding pixels. 9. Pack changed pixel data into packed buffer to ready for output. 10. Output the scan line in the packed buffer to the Detection File. 11. Move up all lines in packed buffer by one line. 12. Move up all lines in the unpacked buffer by one line.	. packed buffer . unpacked buffer

Author: ______________ Date: ______

Diagram ID: 12.14.3.2 Name: ______________ Description: CLASS ANOMALIES

Input	Process	Output
	13. Repeat steps 7 through 12 for all lines up to and including ending line of anomaly boundary. 14. If any other anomalies need processing, repeat steps 1 through 13 for each. 15. When no other anomaly processing is required, output the remainder of the lines in the packed buffer to the Detection File. 16. Read and output remainder of Detection File.	Detection File (copy)

Author:

Date:

Diagram ID: 2.14.4   Name: KMDANO   Description: PROCESS ANOMALIES

Input

Card image

Process

1. Get type of data file containing the anomaly.
2. If type is not CLASS or RADIANCE, write an error message and set error flag.
3. If type of data file specified is not equal to type of Detection File, print an error message and set error flag.
4. If error occurred, return.
5. Get line and sample numbers for all anomalies to be processed.
6. Sort all anomaly line and sample pairs in ascending order.
7. If type of Detection File is RADIANCE, process radiance (2.14.3.1).
8. If type of Detection File is CLASS, process class (2.14.3.2).

Output

. type of data file

Error message

. error flag

. line of anomalies

. sample of anomalies

. number of anomalies

Detection File (copy)

- Radiance
- Class

Author: ______ Date: ______

Diagram ID: 2.14.3.1 Name: ______ Description: RADIANCE ANOMALIES

Input	Process	Output
. spectral neighbor table	8. Read into packed buffer, unpack pixel data, and store in unpacked buffer to provide enough surrounding pixels to equal the radius of closure in all directions for the current anomaly line. 9. For current anomaly line, look at surrounding pixels and using the spectral neighbor table, determine if any anomaly pixels should change their radiance values. If so, determine new values from values of surrounding pixels. 10. Pack unchanged pixel data into packed buffer to be ready for output. 11. Output the scan line in the packed buffer to the Detection File. 12. Move up all lines in packed buffer by one line. 13. Move up all lines in unpacked buffer by one line.	. packed buffer . unpacked buffer

Author: ______ Date: ______

Diagram ID: 2.14.3.1 Name: ______ Description: RADIANCE ANOMALIES

Input

. line of anomaly

. sample of anomaly

. radius table

Detection File

Process

1. Using line and sample location of anomaly pixel, determine the radiance value of that pixel.
2. Use radius table to determine radiance range for the anomaly pixel. Save this radiance range and radius of closure for that pixel.
3. Determine the boundaries of the complete anomaly based on contiguous pixels that falls within the radiance range of the specified anomaly pixel.
4. Using the top boundary of the anomaly and the radius of closure, calculate the starting scan line for saving data outside the anomaly.
5. Save beginning line boundary as current anomaly line.
6. Reinitialize the beginning of Detection File.
7. Read and output all Detection File scan lines (records) that will not change onto Detection File copy.

Output

. beginning line boundary

. ending line boundary

. beginning sample boundaries/line

. ending sample boundaries/line

. starting scan line (STSCAN)

Detection File (copy)

Author: ______ Date: ______

Diagram ID: 2.14.3.1 Name: ______ Description: RADIANCE ANOMALIES

Input	Process	Output
	14. Repeat steps 8 through 13 for all lines up to and including ending line boundary. 15. If any other anomalies to process, repeat steps 1 through 14. 16. When no other anomaly processing is required, output the remainder of the lines in the packed buffer to Detection File. 17. Read and output remainder of the Detection File records.	Detec.ion File (copy)

Author: ____________

Date: 02/26/79

Diagram ID: 2.15 Name: IGS Description: INTERACTIVE GRAPHICS SUBSYSTEM

Input

Graphic/image records (from UNIVAC processing):

- Header record
- Tick marks or axis scaling information
- Graphics/image data

User request (from UNIVAC)

Process

1. Receive and unpack UNIVAC graphics/image records (2.15.1).
2. Perform designated display processing (2.15.2).
3. Transmit to the UNIVAC any legal and required display directive/information (2.15.3).

Output

Graphics/image record on RAMTEK format

Required CRT display:

- Image
- X-Y plot
- Text

Message buffer

Data buffer

Note: This is an overview of the Interactive Graphics Subsystem Functions. This block performs no actual functions as the Interdata and RAMTEK computers are involved. The accomplishment of the above functions

Author:

Date: 02/26/79

Diagram ID: 2.15.1

Name: GINPUT

Description: COMMUNICATION SOFTWARE INTERFACE INPUT (UNIVAC → RAMTEK)

Input

Graphic/image records:

- Header record
- Tick marks or axis scaling information
- Graphics/image data record

Process

1. Receive and decode graphic/image header record information from transmitted UNIVAC header record.
2. Initialize required variable settings.
3. Route input records to appropriate input communication as indicated in the decoded header information.
   a. If the current data is IMAGE display data, then call the Image Display Module (2.15.1.1)
   b. If the current data is X-Y plot Display Module (2.15.1.2).
   c. If the current data is TEXT (tabulation) data, then call the Text Display Module (2.15.1.3).
4. If an error in the header record has been detected, then send a message to user via UNIVAC.

Output

Image CRT display record:

- Radiance
- Gradient
- Class
- Laplacian
- Polar
- Registered

X-Y CRT display record:

- Profile plot
- Histogram

Text CRT display record

UNIVAC error message

Author: ____________ Date: 02/27/79

Diagram ID: 2.15.1.1 Name: INIMAG Description: DECODE UNIVAC IMAGE DATA — PREPARE AND TRANSMIT IMAGE DATA TO RAMTEK

Input

Graphic/image record (UNIVAC format):

- Header record
  - Color table information
  - Tick mark data
- Image data record

Process

1. Initialize required variable settings.
2. Color table information:
   a. Set up color table from information in UNIVAC header record.
   b. Temporarily store color table on a storage medium.
   c. Transmit color table to RAMTEK (2.15.2).
3. Tick mark data:
   a. Set up tick mark table from information in UNIVAC transmission.
   b. Store tick marks on storage medium and transmit tick mark data to RAMTEK (2.15.2).
4. Header and image data:
   a. Reformat input header and image data in UNIVAC format to data in RAMTEK format.
   b. Temporarily store RAMTEK image record on storage medium.

Output

Color table

Tick mark table

Author:

Diagram ID: 2.15.1.1 Name: INIMAG

Date: 02/27/79

Description: DECODE UNIVAC IMAGE DATA — PREPARE AND TRANSMIT IMAGE DATA TO RAMTEK

Input	Process	Output
	c. Transmit image data and header data to RAMTEK (2.15.2).	RAMTEK header Image data in RAMTEK format
	5. If any errors were encountered, output appropriate error message.	UNIVAC error message or RAMTEK CRT error message

Author: ______

Date: 03/01/79

Diagram ID: 2.15.1.2   Name: INPLOT   Description: DECODE UNIVAC PLOT DATA — PREPARE AND TRANSMIT RAMTEK PLOT DATA

Input	Process	Output
Graphic/image records (UNIVAC format): - Header record Color table information Axis scaling information	1. Initialize required variable settings. 2. Color table information: a. Set up color table from information in UNIVAC header record. b. Temporarily save color table on a storage medium. c. Transmit color table to RAMTEK. 3. Alphanumeric annotation for plots: a. Set up alphanumeric plot data from information in UNIVAC header record. b. Transmit character data to RAMTEK. 4. Plot axis and scaling information: a. Set up plot abscissa and ordinate from information in UNIVAC header record.	Color table Plot annotation

Author:

Diagram ID: 2.15.1.2 Name: INPLOT Date: 03/01/79

Description: DECODE UNIVAC PLOT DATA — PREPARE AND TRANSMIT RAMTEK PLOT DATA

Input	Process	Output
Graphic/image records (UNIVAC format): - Graphic data record	b. Set up plot scale marks from information in UNIVAC record. c. Transmit axis-associated information to RAMTEK. 5. Plot points coordinates: a. Set up in a RAMTEK format, plot point X-Y coordinates. b. Transmit X-Y coordinates for all points to be plotted with RAMTEK parameter values set. 6. Repeat step 5 as required for transmission of more than one parameter to be plotted. 7. If an error is encountered, output an appropriate error message.	Plot scales Plot point coordinates UNIVAC error message or RAMTEK CRT error message

Author:

Diagram ID: 2.15.1.3 Name: INTAB Date:

Description: DECODE UNIVAC TEXT DATA – PREPARE AND TRANSMIT RAMTEK TEXT DATA

Input	Process	Output
Graphic/image records (UNIVAC format): - Header record - Text data record	1. Initialize required variable settings. 2. Set up text font information and transmit to RAMTEK. 3. Reformat header and text data in UNIVAC format to data in RAMTEK format. 4. Transmit RAMTEK text data record to RAMTEK (2.15.2). 5. If any errors were encountered, output in appropriate error message.	Text font data Text data UNIVAC error message or RAMTEK CRT error message

Author: ________ Date: 02/26/79

Diagram ID: 2.15.2 Name: CIPROC Description: DISPLAY PROCESSING

Input	Process	Output
Graphics/image record (RAMTEK format)	1. If a graphics/image data record has been transmitted, call the Display Module (2.15.2.1).	Selected CRT display
User request	2. If user has requested via host CPU an action performed (2.15.2.2), then prepare the data buffer and transmit it to the UNIVAC via INTERDATA.	Data buffer for UNIVAC
	3. If any errors are encountered transmit appropriate error message.	Error message

Author:

Date: 02/27/79

Diagram ID: 2.15.2.1 Name: GDISPL Description: DISPLAY GRAPHIC/IMAGE DATA ON CRT

Input	Process	Output
Graphic/image record (RAMTEK format): - Header words - Data words Tick mark table Color table	1. Decode operation code in word 1 of image record and load appropriate registers and perform indicated operation using auxillary information from header words, 2 through 16. 2. If the input record is an image record, then: a. Reset or clear CRT. b. Load memory with tick marks and image data. c. Load color table. d. Write image to CRT. 3. If the input record is a text (tabulation) record, then: a. Reset or clear CRT. Determine font or size of characters to be displayed. b. Load color table. c. Load memory with text data. d. Write tabulations to CRT.	CRT display: - Unregistered images Radiance Gradient Laplacian Rotated Polar Class - Registered images CRT tabulations

Author:

Diagram ID: 2.15.2.1 Name: GDISPL Description: DISPLAY GRAPHIC/IMAGE DATA ON CRT

Date: 02/27/79

Input

Graphic/image record (RAMTEK format):

- Header words
- Data words

Color table

Process

4. If the input record is an X-Y plot record , then:

   a. Reset or clear CRT.

   b. Lcad color table.

   c. Load memory with

      (1) Plot axis.

      (2) Plot annotation.

      (3) X-Y coordinates of points to be plotted.

   d. Write plots (vector connected or point) to CRT.

5. If any errors were encountered, output appropriate error message.

6. If display is temporarily stopped by user, then upon user request the display will continue from point where halted.

7. If the user requests a restart of display, then the display will be output beginning at the first date point.

Output

CRT plots:

- Profile plot
- Histogram plot

UNIVAC error message or RAMTEK CRT error message

Author: ______　　　　Date: ______

Diagram ID: 2.15.2.2　　Name: ______　　Description: PREPARE AND PROCESS HOST CPU REQUESTS

Input

User request for information from host CPU

Process

1. Upon request the host CPU will have the ability to interrogate the following information from the RAMTEK:

   a. Cursor state.

   b. Registers state.

   c. Peripheral status.

   d. Read back current image.

   e. Color table contents.

   f. Auxillary information.

2. If CRT cursor coordinates have been requested, then output this information on CRT.

Output

Requested data

CRT coordinates

Author:

Date: 02/25/79

Diagram ID: 2.15.3 Name: GOUT Description: COMMUNICATION SOFTWARE INTERFACE – OUTPUT (RAMTEK → UNIVAC)

Input	Process	Output
User request	1. Interrogate user request. 2. If a request to display cursor coordinates or a request to collect cursor coordinates is input, then call the cursor point collection module (2.15.3.1). 3. If a color table tabula- tion or a color change is requested, then call the color module (2.15.3.2). 4. If an error is encountered, output the appropriate error message.	Cursor coordinate data buffer: - Current CRT cursor coordinate - Profile point coordinates - Magnification point coordinates for a closed area Color information data buffer: - Color change information - Color table Error message

Author:

Date: 02/28/79

Diagram ID: 2.15.3.1 Name: POINTS Description: PREPARE AND TRANSMIT POINT(S) COORDINATES

Input

User request

User depresses "ENTER" switch

Cursor position

Process

1. Interrogate user request to determine if a point or points has been requested.

2. If user has requested the cursor point coordinates, then:

   a. Store CRT cursor coordinates in RAMTEK memory.

   b. If user has requested the RAMTEK CRT coordinates, then output the RAMTEK coordinates on the CRT.

   c. If user has requested the RAMTEK cursor coordinates to be converted to another coordinate scheme (e.g., degrees), then prepare a data buffer for the UNIVAC containing the CRT coordinates. Transmit this data package to UNIVAC. (Upon conversion the coordinates in the requested coordinate scheme will be output on the alphanumeric terminal.)

Output

RAMTEK CRT coordinate report on CRT

Cursor coordinate data package

Author:

Date: 02/28/79

Diagram ID: 2.15.3.1 Name: POINTS Description: PREPARE AND TRANSMIT POINT(S) COORDINATES

Input	Process	Output
	3. If user has requested that a set of cursor points be collected, then collect all points input from the RAMTEK joystick in the RAMTEK memory. Prepare a data package containing the collected points and transmit the data package to the UNIVAC.	Points coordinate data package
	4. If error, output error message.	UNIVAC error message or RAMTEK error message

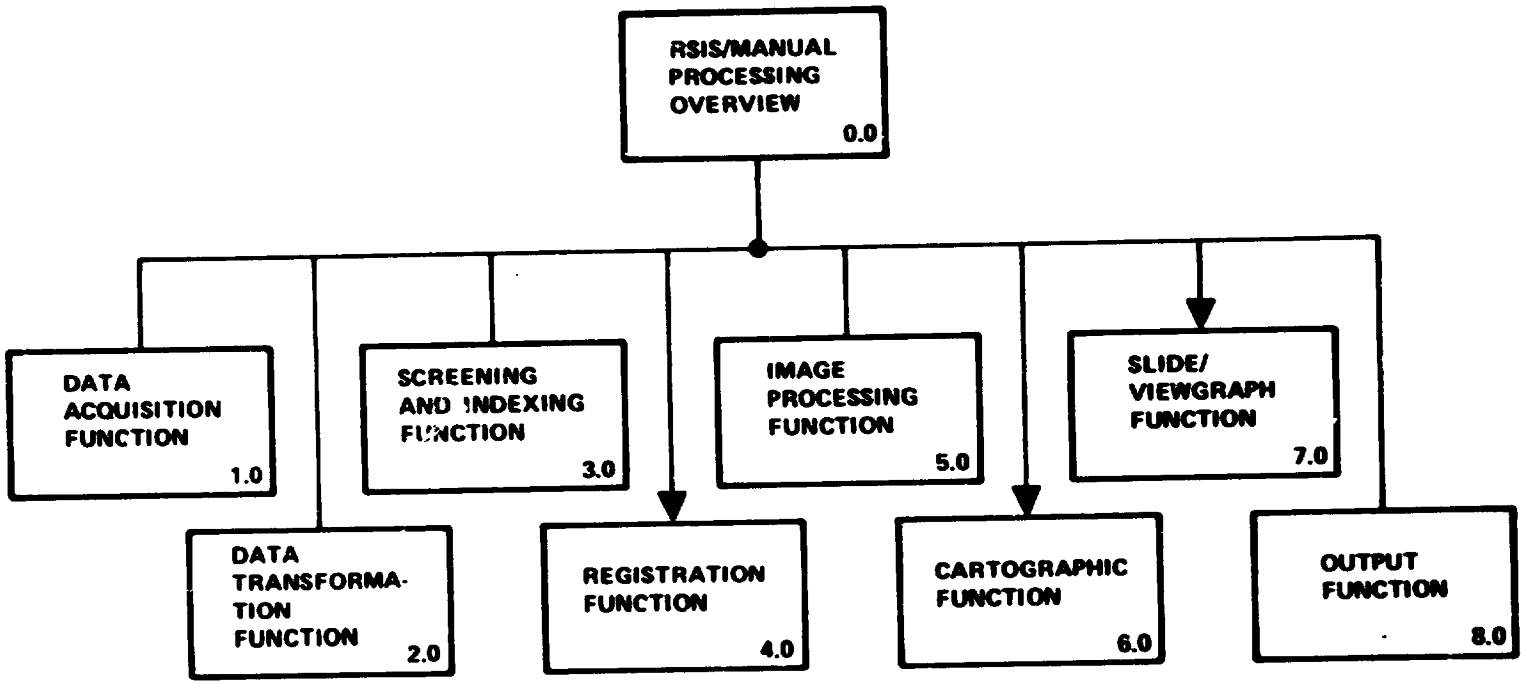
Author: ______

Diagram ID: 2.15.3.2 Name: COLOR

Date: 02/28/79

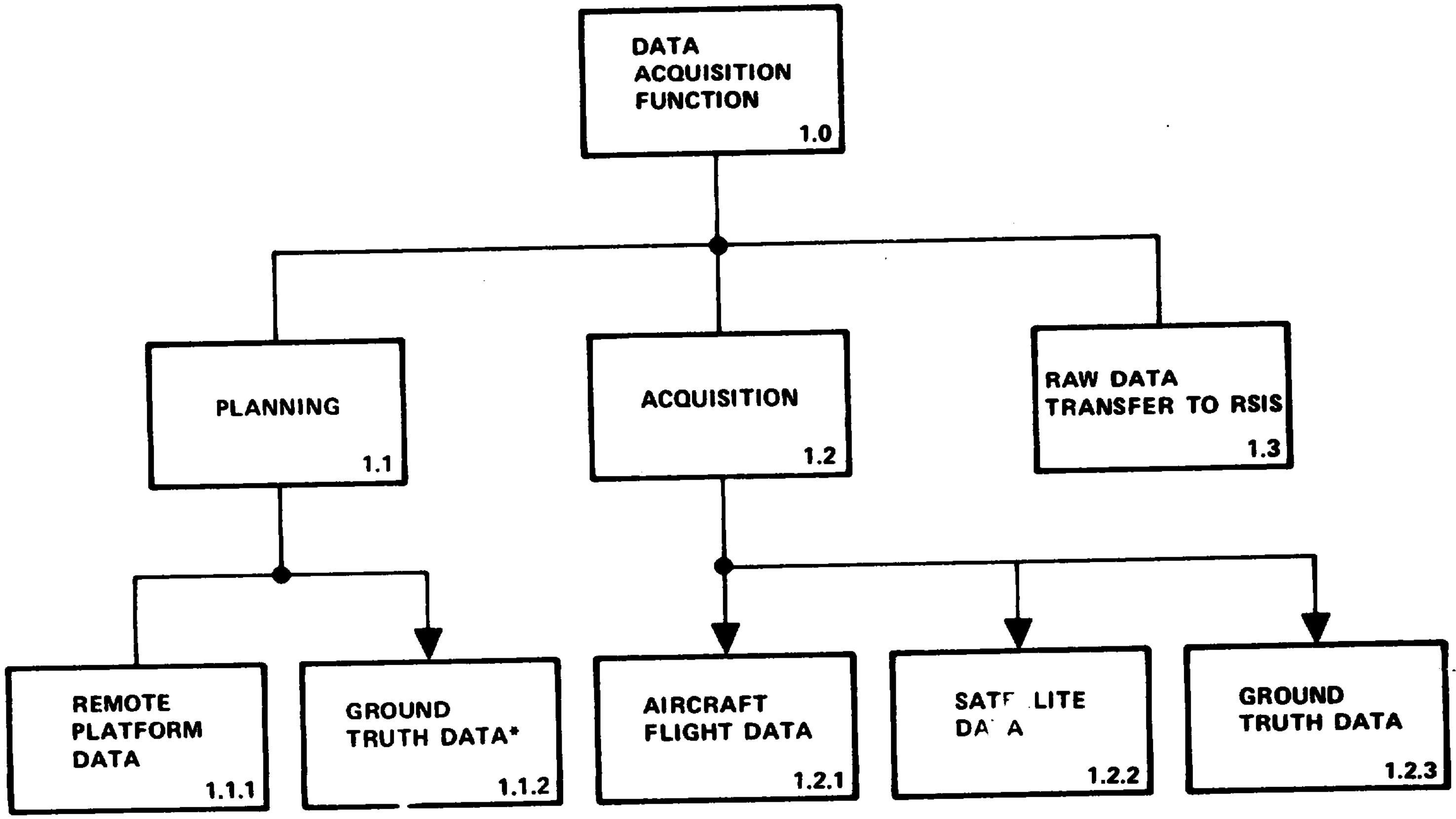
Description: COLOR CHANGE/COLOR TABLE TABULATIONS

Input	Process	Output
User request: - Color change - Color table tabulation	1. Interrogate user request.	
	2. If a color change has been requested by user, then change the appropriate entry on the color table and write to the CRT the current display with color change.	CRT display: - Image - Plot
	3. If a tabulation of the color table has been requested, then display on CRT the color table information. Upon request from user,display previous image or plot.	Color table tabulation
	4. If an error was encountered, output error message.	UNIVAC error message or RAMTEK error message



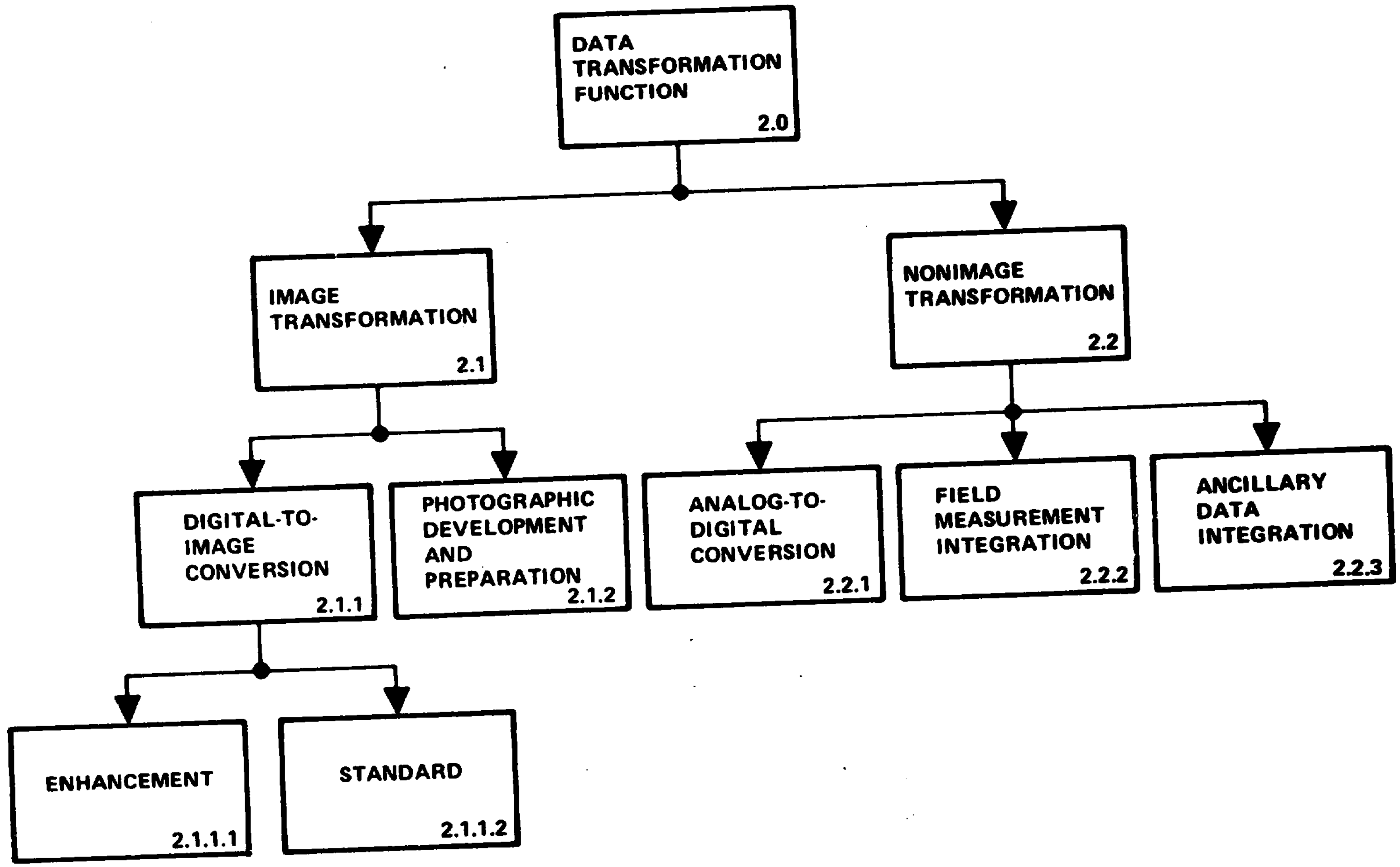
NOTE: ARROWS INDICATE SELECTIVE ROUTING.

Visual table of contents for RSIS manual processing.

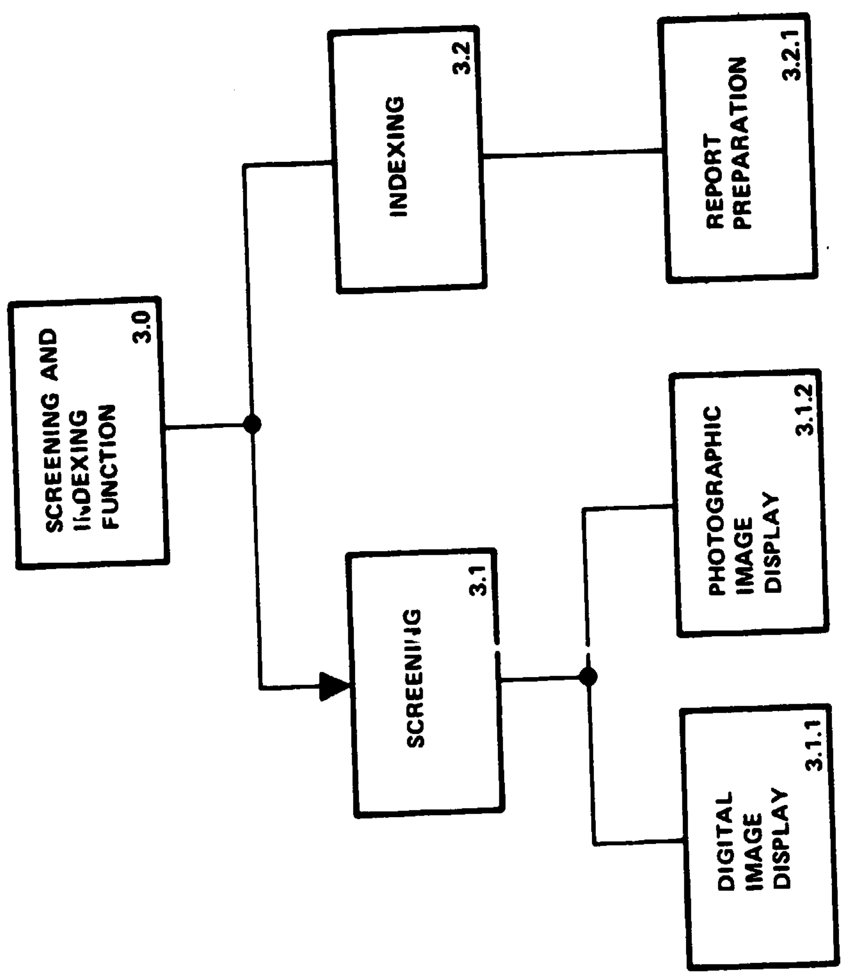


NOTE: ARROWS INDICATE SELECTIVE ROUTING.

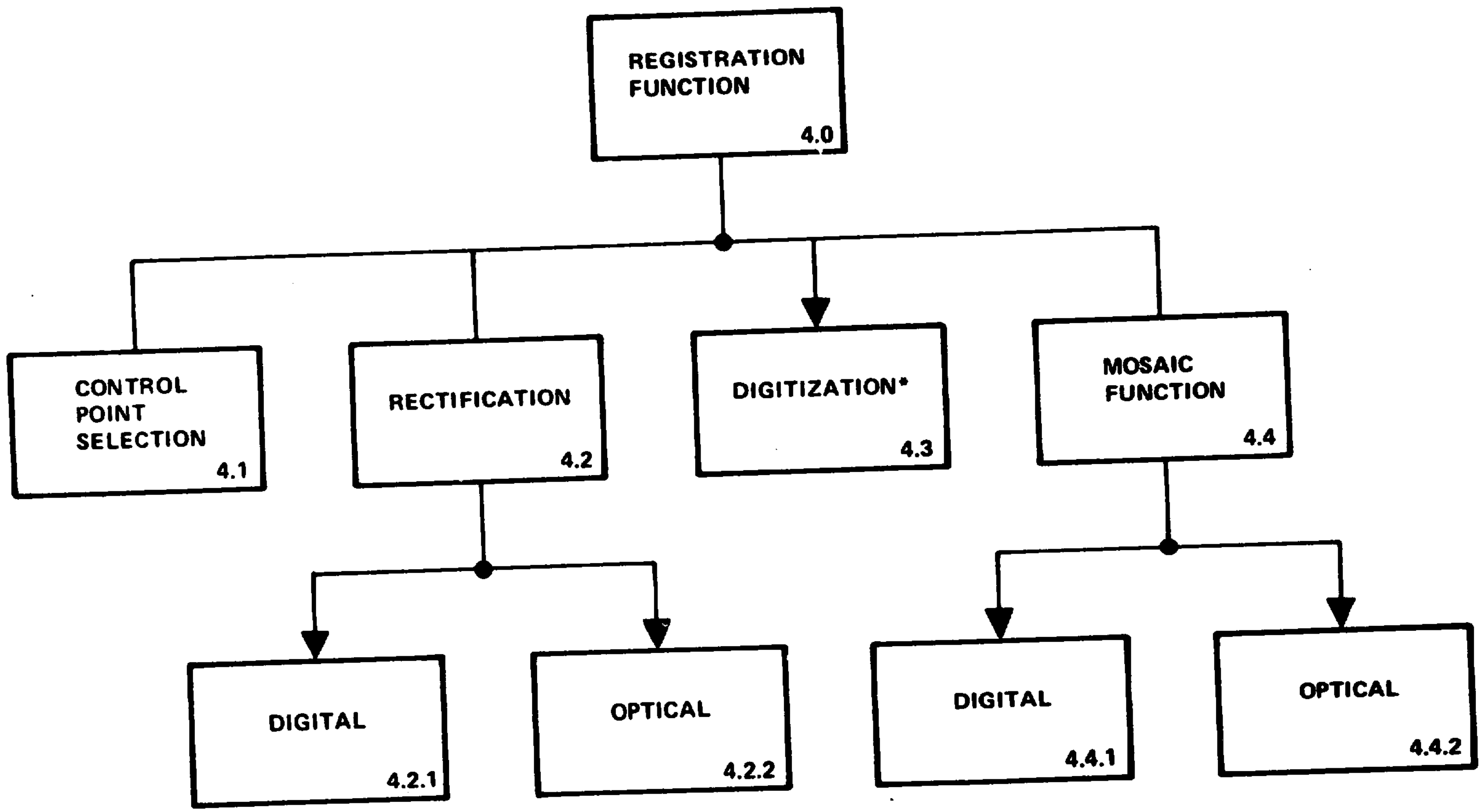
*THE TERM "GROUND TRUTH DATA" IS USED HERE TO REPRESENT FIELD-ACQUIRED DATA AS WELL AS ALL ANCILLARY DATA SUCH AS MAPS, METEOROLOGICAL DATA, WATER SALINITIES, AND SO ON.



NOTE: ARROWS INDICATE SELECTIVE ROUTING.

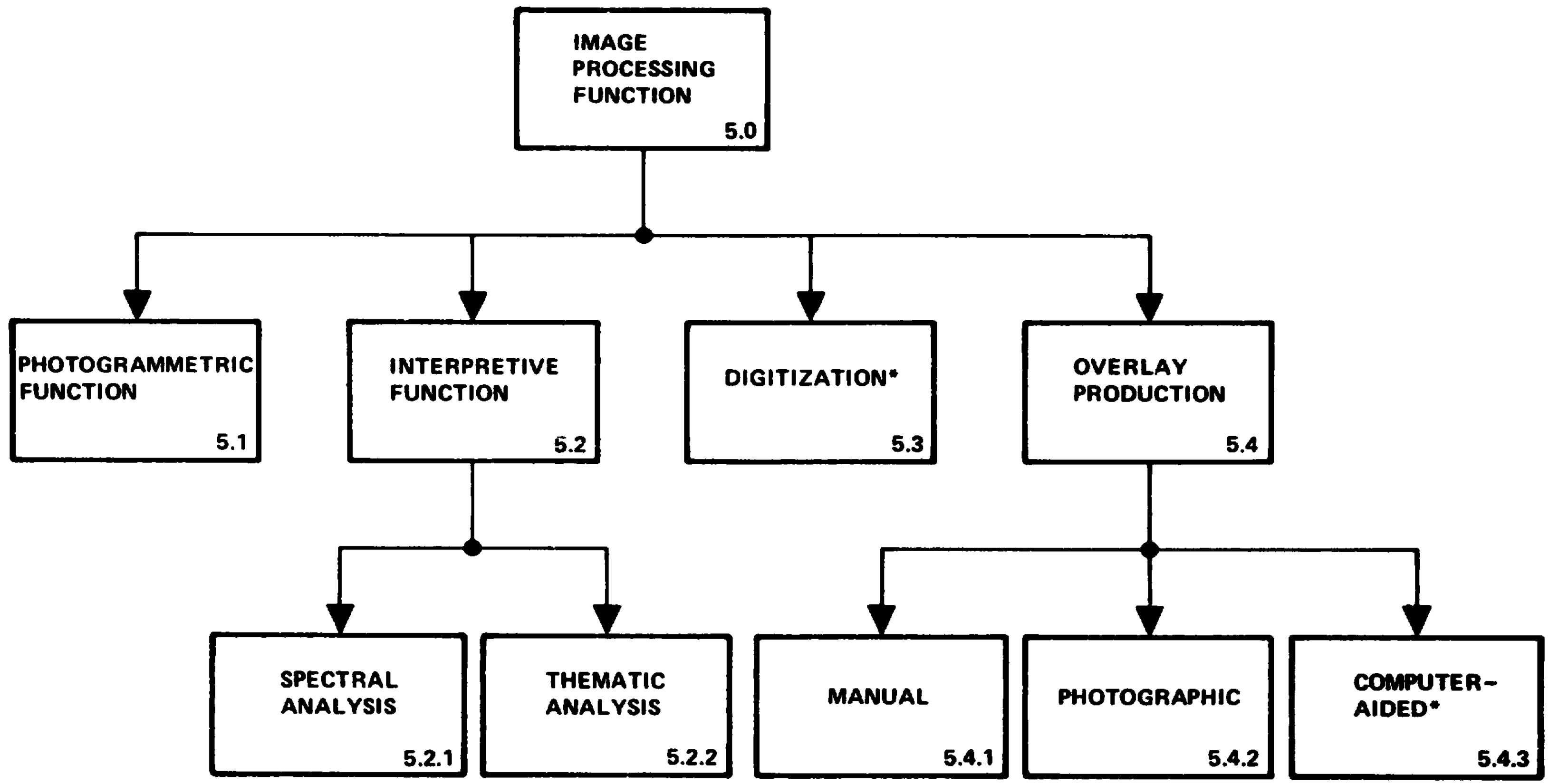


NOTE: ARROWS INDICATE SELECTIVE ROUTING.



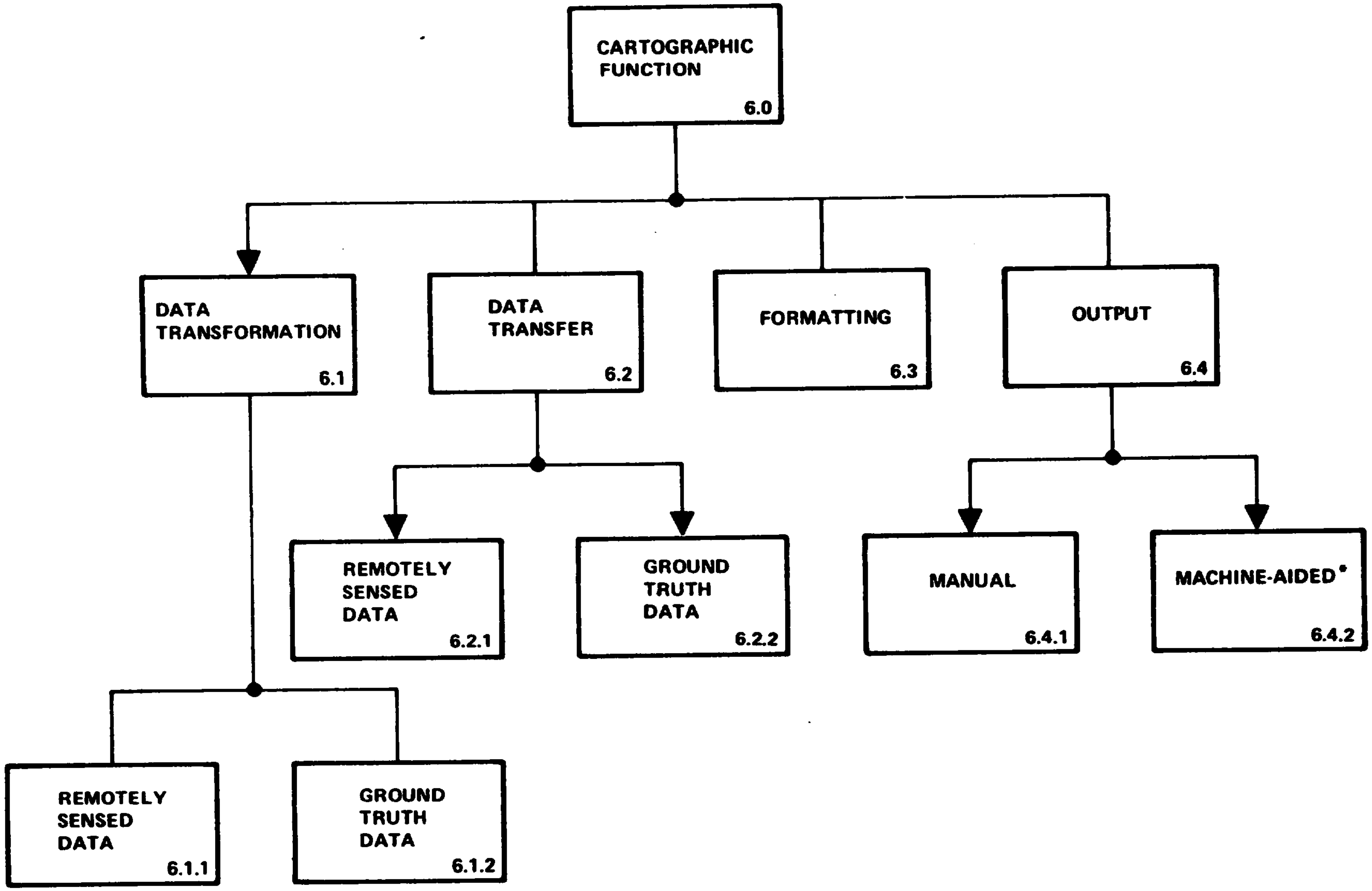
NOTE: ARROWS INDICATE SELECTIVE ROUTING.

*DIGITIZATION FUNCTION PERFORMED IN THE GIS.



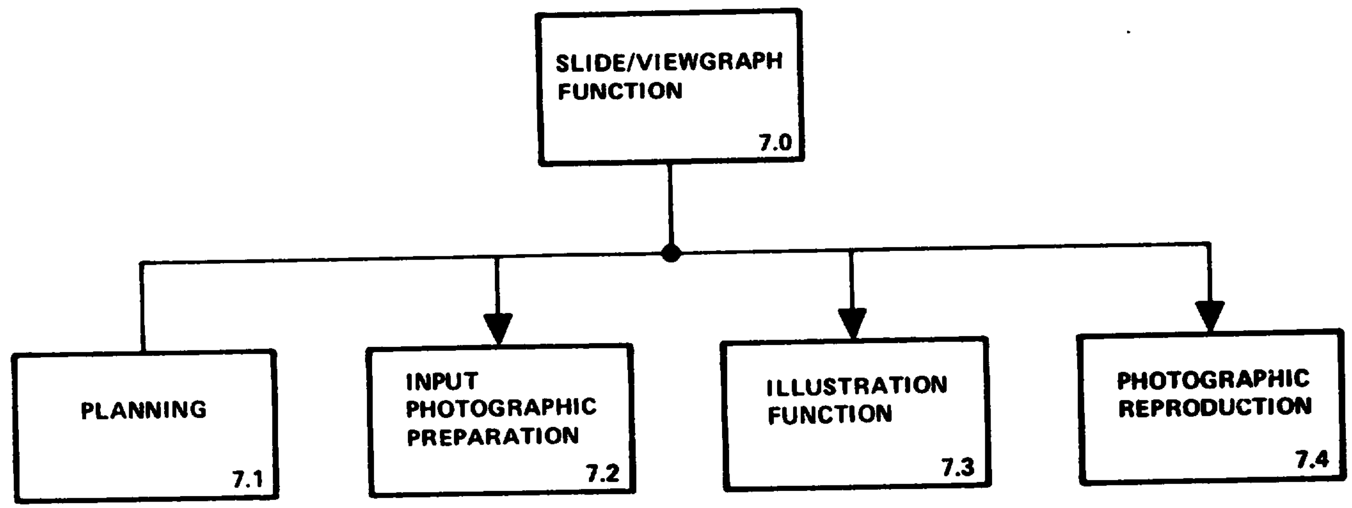
NOTE: ARROWS INDICATE SELECTIVE ROUTING.

*DIGITIZATION FUNCTION PERFORMED IN THE GIS.



NOTE: ARROWS INDICATE SELECTIVE ROUTING.

*DIGITIZED OUTPUT SUPPLIED BY THE GIS.



NOTE: ARROWS INDICATE SELECTIVE ROUTING.

Date:

Author:

Diagram ID: 1.0

Name:

Description: DATA ACQUISITION FUNCTION

Input	Process	Output
Program objectives Term of program (i.e., start-up and end dates) Level-of-effort and funding	1. Plan acquisition activities at ground, aircraft, satellite altitudes. 2. Implement acquisition activities. 3. Transfer acquired data to RSIS and store.	Unprocessed data resides in RSIS facility

Author: ______________

Diagram ID: 1.1 Name: ______________ Date: ______________

Description: PLANNING

Input

Program objectives

Planning materials (maps, photographs, literature, etc.)

Level-of-effort and funding specifications

Data sources

Personnel available

→

Process

1. Identify program requirements.
2. Develop experimental design (depending on objective)
3. Specifically identify study area (layout).
4. Identify possible data sources for remotely sensed data and ground acquired data.
5. Identify gaps in data sources.
6. Schedule tasks in detail.
7. Match personnel with tasks. Identify task leaders.
8. Identify specific data outputs.

→

Output

Work or task objectives

A procedure for data acquisition

Data requirements definition

Generalized map of area(s) under observations.

Specific data sources

Level-of-effort and schedules

Program output by task

Author: ______ Date: ______

Diagram ID: 1.1.1 Name: ______ Description: REMOTE PLATFORM DATA

Input	Process	Output
Platforms and sensors available	1. Match platform and sensor* with data needs.	Flight plans
Schedules of availability	2. Plan overflight and alternative times and dates.	Schedules of data acquisition
Maps and photographs of study site	3. Specifically locate flight paths and altitudes on maps.	Determination of alternative data sources
	4. Consider temporal characteristics (season, time of day, sensor on/off time, etc.) and ground data collection plan.	
	5. Assess previously gathered data for possible use.	

*"Sensor" used in this context refers to hardware type as well as film/filter/lens combinations, scan rates, etc.

Author: ______

Diagram ID: 1.1.2 (optional) Name: ______ Date: ______

Description: GROUND TRUTH DATA

Input	Process	Output
Existing ground truth data including maps, photographs, vegetation keys, soils and climatic data, etc. Data collection devices Names of landowners, agencies, organizations who can give permission to trespass.	1. Reconnaissance of study area on ground or by photograph. 2. Selection of specific study locations. 3. Acquisition of ground truth data collection devices (cameras, boats, etc.). 4. Obtain permission to trespass. 5. Secure transportation, housing, subsistance, funding. 6. Develop field guide routing, points of data acquisition, and schedules. 7. Identify participants. 8. Develop method of field data storage (i.e., plastic bags, photographs, soil cores, etc.) and sample-identification technique.	Total field guide including: a. Routing b. Sampling points c. Method of sampling d. Schedules e. Participants Permission to enter sample locations Transportation, housing, subsistance, and funding arrangements

Author: ______ Date: ______

Diagram ID: 1.2 Name: ______ Description: ACQUISITION

Input

Data acquisition plans and field guide

Acquisition hardware (i.e., ground truth data collection devices)

Process

1. Request for overflight data procurements, ground truthing devices.
2. Obtain approval for expenditures of funds.
3. Transfer relevant information to flight control, ground truth manager, etc.
4. Implement schedules for all data collection.
5. Monitor acquisition, coordinate simultaneous activities (overflight and ground truth).
6. Preprocess data if required.
7. Acquire unprocessed or processed data and supplementary information such as digital data formats.

Output

Raw data, satellite aircraft, ground truth

Supplemental information

Procedure for receipt of continuous update if required

Author: ______ Date: ______

Diagram ID: 1.2.1 Name: ______ Description: AIRCRAFT FLIGHT DATA

Input	Process	Output
Flight paths Schedules Alternative flight plans and schedules Sensors required	1. Prepare aircraft (sensors, personnel, etc.). 2. Implement schedules. 3. Fly data runs. 4. Monitor flight. 5. Acquire unprocessed data.	Unprocessed data

Author: ______

Date: ______

Diagram ID: 1.2.2

Name: ______

Description: SATELLITE DATA

Input	Process	Output
Satellite flight data (Landsat, Seasat, weather, etc.) Procurement forms Aircraft and ground truthing schedules	1. Review flight data (dates, quality frequency, etc.). 2. Procure data from appropriate agency. 3. Coordinate aircraft and ground truth schedules. 4. Procure operations format to utilize data. 5. Preprocess data if required at appropriate agency.	Satellite images and/or magnetic tape data Operations format

Author: ______

Diagram ID: 1.2.3

Name: ______

Date: ______

Description: GROUND TRUTH DATA

Input

Field guide

Logistical arrangements

Letters of permission to trespass

Data collection stations

Process

1. Review plan with participants.
2. Implement schedules and activities.
3. Secure ground truth data required.
4. Prepare alternate data collection arrangements when required.
5. Establish continuous recording DCS.
6. Test DCS.
7. Prepare data output.
8. Interface with local state and federal agents.

Output

Ground truth data identified by specific location.

Helpful contacts in state and federal agencies

Author: ______ Date: ______

Diagram ID: 1.3 Name: ______ Description: RAW DATA TRANSFER TO RSIS

Input	Process	Output
Unprocessed or preprocessed data	1. Screen data for completeness. 2. Physical transfer to RSIS facility. 3. Implement procedure for continuous update (e.g., Landsat DCS).	Unprocessed data in RSIS facility in digital or image format

Author: ______ Date: ______

Diagram ID: 2.0 Name: ______ Description: DATA TRANSFORMATION FUNCTION

Input

Unprocessed or preprocessed data

Transformation hardware and software

→

Process

1. Alter unprocessed data to format where information can be obtained.
2. Integrate data with field measurements or ancillary data if required.
3. Enhance data such that information is easily extracted.

→

Output

Image and/or nonimage products

Author: ______ Date: ______

Diagram ID: 2.1 Name: ______ Description: IMAGE TRANSFORMATION

Input

Digital tape

Digital image, undeveloped product

Undeveloped film products

Encoding technique

Process

1. Digital to image conversion (see computer processing, 2.12), single or multiple channels.
2. Process film products.
3. Change scales.
4. Encode data if required.

Output

Image of study areas with encoded information

Author: ______ Date: ______

Diagram ID: 2.1.1 Name: ______ Description: DIGITAL TO IMAGE CONVERSION

Input

Process

(See Computer Processing Section, 2.12).

Output

Author: ______ Date: ______

Diagram ID: 2.1.1.1 Name: ______ Description: ENHANCEMENT

Input

Program objectives

Enhancement software

Process

1. Enhance for clarity (i.e., resolution).
2. Enhance for spectral discrimination.
3. Enhance a single level or multiple spectral levels.
4. Correct for atmospheric and/or sun angle variations.
5. Prepare composite image (multiple images; multi-date, aircraft and satellite, multi-polarized, radar and satellite, etc.).

Output

Enhanced image

Date: ______

Author: ______

Diagram ID: 2.1.1.2 Name: ______ Description: STANDARD

Input	Process	Output
Established criteria for image transform	1. Convert data to image format following established criteria set.	Standard output product

Author: __________ Date: __________

Diagram ID: 2.1.2 Name: __________ Description: PHOTOGRAPHIC DEVELOPMENT AND PREPARATIONS

Input		Process		Output
Undeveloped film products	→	1. Develop data in accordance with standard procedures. 2. Develop data in accordance with some special procedures. 3. Change scales. 4. Merge with other data (e.g., overlays, control point information, coordinates, etc.).	→	Standard photographic image Special photographic image

Author: ______________________ Date: __________

Diagram ID: 2.2 Name: ______________________ Description: NONIMAGE (DATA) TRANSFORMATION

Input	Process	Output
Nonimage data Storage and retrieval software Transformation algorithms Encoding technique	1. Store single or multiple data sets. 2. Transform data (i.e., atmospheric correction, sun angle correction, reduction of superfluous noise, etc.). 3. Encode data. 4. Recall single or multiple data sets.	Digital data output of single or multiple data sets Transformed data

Date: ______

Author: ______

Description: ANALOG-TO-DIGITAL CONVERSION

Diagram ID: 2.2.1   Name: ______

Input

Analog data (maps, photographs, transects, etc.)

Digitized sampling rate

Digital image data

Storage and retrieval software

Digitizing hardware

Encoding technique

→

Process

1. Sample data at predetermined rate.
2. Correlate sampling on all data sets.
3. Encode data cells, specific points or areas.
4. Store single or multiple data sets.
5. Recall single or multiple data sets.

→

Output

Digital data output of single or multiple data sets correlated to each other and encoded

Author: ______ Date: ______

Diagram ID: 2.2.2 Name: ______ Description: FIELD MEASUREMENT INTEGRATION

Input

Field data

Reduction algorithms

Storage and retrieval software

Process

1. Correlate field data with ancillary and remotely sensed data.
2. Reduce field measurements to usable format.
3. Sample data at predetermined rate if required.
4. Store single or multiple data sets.
5. Encode data for 1.0 purposes.
6. Recall single or multiple data sets.

Output

Field data correlated with remotely sensed and/or ancillary data

Author: ______ Date: ______

Diagram ID: 2.2.3 Name: ______ Description: ANCILLARY DATA INTEGRATION

Input

Ancillary data

Reduction algorithms

Storage and retrieval software

Process

1. Correlate data with field and remotely sensed data.
2. Reduce ancillary data.
3. Sample data at predetermined rate if required.
4. Encode data for 1.0 purposes.
5. Store single or multiple data sets.
6. Recall single or multiple data sets.

Output

Ancillary data correlated with field data and/or remotely sensed data and encoded if required

Author: ______________ Date: ______

Diagram ID: 3.0 Name: ______________ Description: SCREENING AND INDEXING

Input	Process	Output
Transformed image data Transformed nonimage data Maps	1. Review all data. 2. Index data on a map. 3. Prepare a report on data quality, usability, coverage, gaps, etc.	Data accepted for program Request for supplemental data

Author: ______ Date: ______

Diagram ID: 3.1 Name: ______ Description: SCREENING

Input

All transformed image or nonimage data

→

Process

1. Display all digital data.
2. Display all photographic data.
3. Review data for completeness, quality, coverage, usability, correlation of data sets.

→

Output

Accept/reject decision capability

Request for additional or different data to satisfy program needs

C.5

Author: ______ Date: ______

Diagram ID: 3.1.1 Name: ______ Description: DIGITAL IMAGE DISPLAY

Input

Display image

Digital CCT data

→

Process

1. Display all digital data and screen.
2. Produce hardcopy output if required.

→

Output

Visual review of data

Hardcopy output

Author:

Date:

Diagram ID: 3.1.2

Name:

Description: PHOTOGRAPHIC IMAGE DISPLAY

Input	Process	Output
Photographic data	1. Display all data and screen. 2. Produce hardcopy.	Visual review of data Hardcopy output

Author: ____________ Date: ____________

Diagram ID: 4.0 Name: ____________ Description: REGISTRATION FUNCTION

Input	Process	Output
Ground or photograph reference base Remotely sensed data Registration hardware or software	1. Establish control. 2. Rectify or digitize data. 3. Mosaic data if required. 4. Execute registration.	Remotely sensed data and/or reference data in registration

Author: ______ Date: ______

Diagram ID: 4.1 Name: ______ Description: CONTROL POINT SELECTION

Input	Process	Output
Ground or photograph reference data Data to be registered to reference	1. Select potential control points on reference data. 2. Estimate coordinate position of control points on data to be registered. 3. Digitize positional data if required (i.e., 4.3).	The estimated coordinates of control points on reference data and data to be registered

Author: ______ Date: ______

Diagram ID: 4.2 Name: ______ Description: RECTIFICATION

Input

Digital or analog data

Rectification model

Potential control points

Process

1. Least squares fit of raw input data to a reference set of control points.

Output

Best fit of remotely sensed data to control point

Author: ____________ Date: ____________

Diagram ID: 4.3 Name: ____________ Description: DIGITIZATION

Input	Process	Output
Analog data Digital sample rate Digitizing hardware and software	1. Digitize data. 2. Store digitized data.	Data stcred in digital format on CCT.

Author: ______ Date: ______

Diagram ID: 4.4 Name: ______ Description: MOSAIC FUNCTION

Input

Rectified data or unrectified data

Process

1. Perform manual mosaic of rectified or unrectified data.
2. Reduce scale optically if required.
3. Perform mosaic of digital data and store.
4. Produce hardcopy output if required.

Output

Mosaic print

Mosaic data stored on CCT

Author: ____________ Date: ____________

Diagram ID: 4.5 Name: ____________ Description: EXECUTE REGISTRATION

Input	Process	Output
4.1, 4.2, 4.3, or 4.4 output Registration hardware and/or software	1. Execute optical or computer aided registration.	Registered data sets registered to each other or reference grid

Author: ____________

Date: ____________

Diagram ID: 5.0

Name: ____________

Description: IMAGE PROCESSING FUNCTION

Input	Process	Output
Image or digital products	1. Perform photogrammetric measurements. 2. Perform interpretation of data. 3. Digitize measurements or interpretations. 4. Produce map overlays.	Map information

Author: ____________ Date: ____________

Diagram ID: 5.1 Name: ____________ Description: PHOTOGRAMMETRIC FUNCTION

Input	Process	Output
Rectified, registered or mosaiced data Digital data	1. Perform measurements. 2. Compute areas from digital data. 3. Perform digitization (5.3) for computer-aided computations or overlay production.	Geometric measurements

Author: ______

Diagram ID: 5.2 Name: ______ Date: ______

Description: INTERPRETIVE FUNCTION

Input

Rectified, registered or mosaiced data

Digital data in image form

→

Process

1. Perform spectral analysis of scene.
2. Perform thematic analysis.

→

Output

Subjective interpretation of image data

Author: ____

Diagram ID: 5.3 Name: ____ Date: ____ Description: DIGITIZATION

Input

Interpretations of spectral or thematic analyses

Photogrammetric data

→

Process

1. Digitize interpreted results.
2. Digitize geometric characteristics.

→

Output

CCT of interpretations and/or geometric characteristics

Author: ____________ Date: ____________

Diagram ID: 5.4 Name: ____________ Description: OVERLAY PRODUCTION

Input	Process	Output
Photogrammetric measurements Interpretations (raw data)	1. Manually reproduce interpretations. 2. Photographically reproduce interpretations. 3. Process CCT's to reproduce computer plots of interpretations.	Interpretive overlays and quantitative measurements of geometric characteristics

Author: ____________ Date: ________

Diagram ID: 6.1 Name: ____________ Description: DATA TRANSFORMATION

Input	Process	Output
Processed remotely sensed data Processed ground truth data	1. Refine data for final product. 2. Perform statistical analyses. 3. Reduce data. 4. Summarize data.	Data prepared for map on overlay product

Author: ______________ Date: ______

Diagram ID: 6.2 Name: ______________ Description: DATA TRANSFER

Input

Transformed data

Process

1. Manually transfer data to final product.
2. Optically transfer data to final product.

Output

Data displayed on final product at final or working scales

Author: ______

Diagram ID: 6.3 Name: ______ Date: ______

Description: FORMATTING

Input	Process	Output
All technical input to final product	1. Prepare legends and borders. 2. Prepare final product prototype. 3. Final review of product.	Prototype of final (map) product

Author: ______________________ Date: __________

Diagram ID: 6.4 Name: ______________________ Description: OUTPUT

Input

Prototype of final product

→

Process

1. Manually reproduce maps.
2. Reproduce maps using machine-aided techniques.

→

Output

Final product